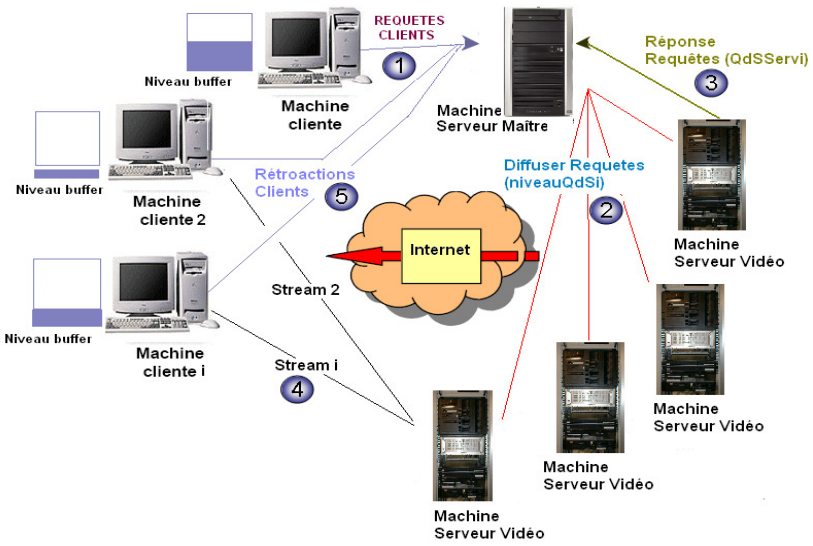
UNIVERSITÉ DU HAVRE – UFR SCIENCES ET TECHNIQUES

Projet tutoré

Modélisation, conception et réalisation d'un simulateur de systèmes multimédias distribués



GAFFÉ Yoann
Master I Informatique

Encadrant : M. DUVALLET

Table des Matières

INTRODUCTION **3**

CHAPITRE 1 : LES VOD **4**

1.1 PRESENTATION 4
1.2 PROBLEMATIQUE : LA QUALITE DE SERVICE (QDS) 4

CHAPITRE 2 : SIMULATION **6**

2.1 PRESENTATION DE LA SIMULATION 6
2.2 CONCEPTION 7
2.2.a Le package Topologie *7*
2.2.b Le package Transformation *12*
2.2.c Le package IHM *14*
2.2.d Le package Graphes *19*

CHAPITRE 3 : MANUEL D'UTILISATION **20**

3.1 LA BARRE DES MENUS 20
3.1.a Le menu Fichier *20*
3.1.b Le menu Insertion *21*
3.1.c Le menu Modifier *23*
3.1.d Le menu Demandes *24*
3.1.e Le menu Aide *25*
3.2 LA BARRE DES BOUTONS 25
3.3 LES ONGLETS 26
3.3.a L'onglet graphe *26*
3.3.b L'onglet desrciption *26*

CHAPITRE 4 : CONCLUSION **28**

4.1 RESULTATS 28
4.2 AMELIORATIONS 34

ANNEXES **35**

ANNEXE 1 : EXEMPLE DE FICHIER .SIM 35
ANNEXE 2 : EXEMPLE DE FICHIER .VIS. 37

Liste des Figures

Figure 1. Scénario de fonction de l'architecture FCS-MS : 5
Figure 2. Présentation de la simulation 6
Figure 3. Schéma UML : Topologie 1 9
Figure 4. Schéma UML : Topologie 2 10
Figure 5. Schéma UML : Topologie 3 11
Figure 6. Schéma UML : Topologie 4 12
Figure 7. Schéma UML : Transformation 13
Figure 8. Schéma UML : Ihm 18
Figure 9. Schéma UML : Graphe 19

Introduction

De nombreuses applications multimédias sont apparues sur internet tout au long de son développement. En quoi consistent ces applications? Généralement, il s'agit de flux de données sous formats texte, son, image et graphique animé. Pour ce projet de master, il nous a été proposé de modéliser, concevoir et réaliser un simulateur de systèmes multimédias distribués.

Ces systèmes sont des systèmes dont les informations stockées dans une base de données multimédias peuvent être divisées en deux parties : une information multimédia (texte, image, son, vidéo) stockée et consultée par l'application, et une information de contrôle (scénarios de synchronisation, paramètres de qualité de service, …). Il s'agissait ici d'implémenter un programme permettant de simuler un bon exemple de ces systèmes avec la vidéo à la demande tout en nous appuyant sur des travaux effectués par une étudiante de master recherche à l'université du Havre s'inspirant d'un autre travail effectué par une étudiante suédoise.

Nous nous intéresserons dans un premier temps à présenter succinctement la VoD et à observer la problématique qu'elle propose pour la simulation. Ensuite, nous tenterons de présenter la simulation et d'expliquer sa conception. Nous construirons alors un manuel d'utilisation. Et enfin, nous observerons les résultats obtenus et les améliorations possibles.

CHAPITRE 1 :

La VoD (Video on Demand)

1.1. Présentation

La vidéo à la demande (VoD) de l'anglais *Video on Demand* est une technique de diffusion de contenus vidéos numériques bidirectionnelle (interactive) offerts ou vendus par les réseaux câblés, comme Internet, ou les réseaux non câblés comme la téléphonie 3G.

Le principe est simple, un utilisateur connecté au réseau souhaite regarder un film. A l'aide d'un décodeur numérique, l'utilisateur a alors accès à une liste de films disponibles sur un serveur vidéo. Il commande son film et quelques secondes ou quelques minutes plus tard, il peut visualiser son film en temps réel son film avec les mêmes fonctions qu'un magnétoscope (lecture, avance rapide, retour en arrière).

1.2. Problématique : la qualité de service (QdS)

Le terme qualité de service désigne la capacité à fournir un service conforme à des exigences en matière de temps de réponse et de bande passante. Celle-ci est un facteur très important dans la VoD car l'utilisateur exige généralement une qualité de service lui permettant de visionner un film sans perte d'images ou perte de données. En effet, pour que l'utilisateur puisse avoir une bonne qualité, il faut prendre en compte la congestion du réseau et la perte de paquets.

Le problème est donc de trouver un système pour pallier à ces soucis de perte de données. Dans son rapport sur la gestion de la qualité de service, Melle Besma Zeddini propose un scénario pour répondre à ce problème, c'est celui-ci que nous avons essayé d'implémenter et de visualiser à travers ce projet.

Scénario proposé par Melle Besma Zeddini :

1. L'ensemble des clients initialisent le déroulement de l'application par le lancement des requêtes. Les requêtes seront alors placées dans une file d'attente avant d'être traitées par le contrôleur d'admission selon la méthode FIFO.

2. Le contrôleur d'admission agit ensuite en fonction de deux paramètres :

- Le nombre de client maximum
- Le niveau de qualité de service maximum

Il commencera par vérifier le niveau du premier paramètre.

- S'il est satisfaisant alors cette requête est envoyée à l'ordonnanceur puis au contrôleur de QdS du serveur maître.

3. Le serveur maître fait suivre la requête au contrôleur d'admission du premier serveur vidéo.

4. Le contrôleur d'admission compare le niveau de QdS demandé par le client par rapport au niveau de QdS du serveur vidéo « i ».

5. Dans le cas où la contrainte est vérifiée, le contrôleur d'admission du serveur vidéo informe le serveur maître de l'acceptation. Ce dernier informe le client sur le serveur vidéo qui prendra en charge l'exécution de sa requête.

6. La requête sera mise dans la file des requêtes bloquées dans deux cas :

- Si initialement le paramètre « nombre de client max » est dépassé
- Si le niveau de QdS du serveur « i » est insuffisant

La réactivation de la requête interviendra si le contrôleur d'admission modifie les paramètres, sinon elle pourra être abandonnée dans le cas où le client annule sa demande.

7. L'exécution de la requête s'accomplira au niveau de serveur vidéo désigné, celui-ci envoie les flux vidéo au client « i », qui seront stockés dans le buffer avant d'être exploités. Suite à ceci, le client établie un jugement (QdS-humain), et envoi un feedback au serveur maître.

8. Les feedbacks arrivent au niveau du moniteur et seront collectés, puis passé à l'ordonnanceur. Son rôle consistera à ordonnancer les feedbacks entrant en fonction de la priorité exprimée pour l'ensemble de client.
Le client le plus prioritaire, sera traité par le contrôleur de QdS qui ajuste le niveau de QdS.

9. Le serveur maître envoie au serveur vidéo « i » le paramètre mise à jour. La redistribution des flux vidéo s'effectuera en prenant en considération la valeur du paramètre ajusté suite à la rétroaction des clients.

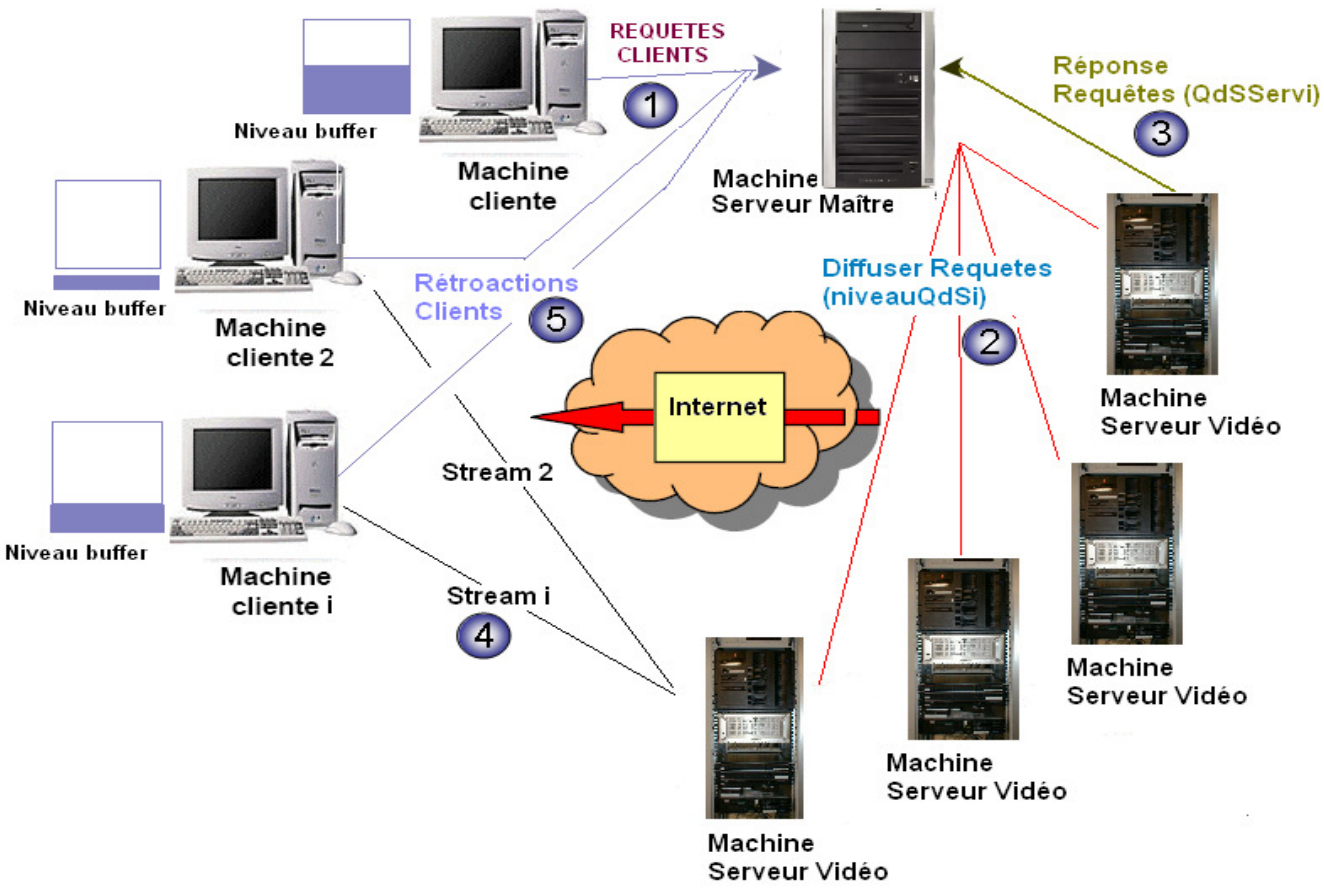


Figure 1. Scénario de fonction de l'architecture FCS-MS
(proposé par Melle Besma Zeddini)

CHAPITRE 2 :
Simulation

2.1. Présentation de la simulation

La simulation se fait sous la forme d'un programme contenant une interface graphique permettant à l'utilisateur de modifier les paramètres du VoD. En effet, ce programme peut lui permettre d'ajouter ou de supprimer des serveurs clients ou des serveurs vidéos du système, ainsi que de les modifier c'est-à-dire de changer la taille du buffer des serveurs clients, d'ajouter un film sur un serveur vidéo ou de modifier sa qualité de service (QdS) maximum. Il lui est aussi possible de créer des demandes simples ou des demandes simultanées de films de la part de serveur client en paramétrant le niveau QdS demandé. Pour cette simulation, nous nous limitons à 7 serveurs vidéo et clients, et nous admettons que le nombre de demandes auquelles un serveur vidéo peut répondre simultanément est de 2 pour rester dans les proportions.

L'utilisateur peut de plus visualiser les événements créés sur le système à un instant t donné, grâce à un timer introduit dans le programme.

Tout d'abord sous forme graphique, où les différents serveurs sont représentés ainsi que la congestion du réseau. Les différents flux de données sont représentés par des couleurs afin que l'utilisateur puisse observer quel est l'événement qui se produit à l'instant t.

Ensuite sous forme écrite, ce qui lui permet de savoir les événements qui se sont produits sur le système et les transferts en cours à un instant t.

Enfin, l'utilisateur a la possibilité de sauvegarder sa simulation ou d'ouvrir une simulation existante.

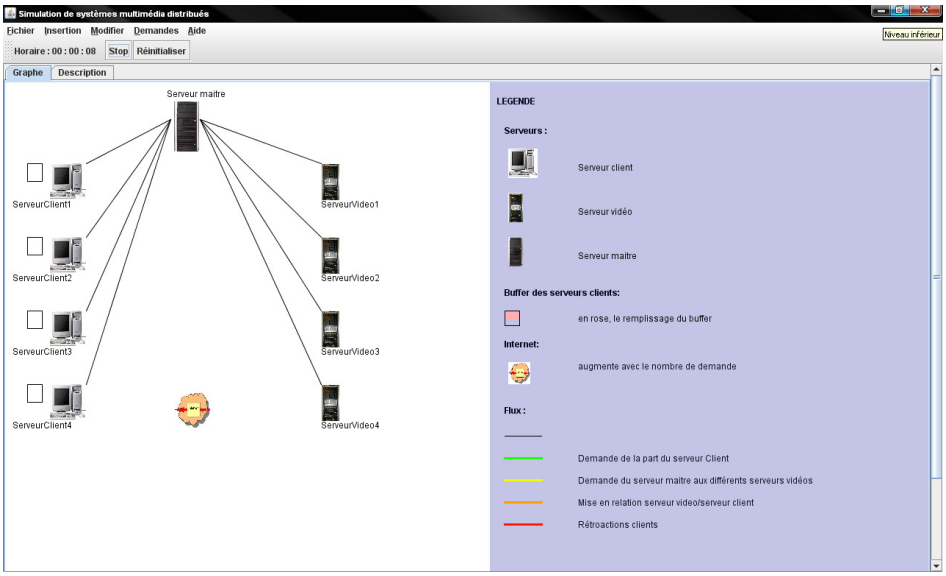


Figure 2. Présentation de la simulation

2.2. Conception

Le programme est conçu sous formes de quatre packages, afin de bien différencier le rôle de chacun d'eux et de faciliter la reprise de ce programme pour des améliorations :

- package Topologie : représentation de tous les objets constituant un système multimédias distribués (serveurs clients, serveurs vidéos, demandes, serveur maître …).
- package Transformation : permet la modification des données sur le système multimédia.
- package IHM : tout ce qui est interface graphique et gestion d'événements dans la fenêtre de l'application
- package Graphe : pour donner à notre système une structure de graphe, afin d'appliquer d'éventuelles améliorations au programme.

2.2.a. Package Topologie :

Le package Topologie est surement le package le plus important du programme. En effet, c'est dans ce package que nous retrouvons les classes représentant pratiquement tous nos objets constituant le système multimédia.

Nous essaierons par la suite d'expliquer la fonction des principales classes et de présenter les méthodes de celles-ci.

▪ Les serveurs :

Les serveurs clients et vidéos seront des objets qui hériteront de la classe Noeud du package Graphes que nous présenterons par la suite. Essayons à présent de les observer plus en profondeur.

ServeurClient :

Présentation des attributs :

- attributs principaux :
  - *nom* : son nom
  - *tailleBuffer* : la taille de son buffer.

- autres attributs :
  - ses coordonnées x et y dans la représentation graphique du système.
  - des attributs booléens *demande* et *retroaction* pour savoir si un serveur client demande un film au serveur maître et si celle-ci est une rétroaction.
  - un attribut *tailleBufferDemande* qui correspond en fait à la taille qui est utilisée dans le buffer d'un serveur client lorsque celui-ci reçoit un film d'un serveur vidéo.

Présentation des méthodes :

- constructeur par défaut et constructeur avec paramètres (son nom et la taille du buffer).
- accesseurs et modifieurs des différents attributs.

ServeurVideo :

Présentation des attributs :

- attributs principaux :
  - *nomServeur* : son nom
  - *niveauQds* : le niveau maximum de QdS que le serveur peut fournir.
  - *nbFilm* : le nombre de films présents sur le serveur
  - *lesFilms* : liste des films présents sur le serveur

- autres attributs :
  - ses coordonnées x et y dans la représentation graphique du système.
  - l'attribut booléen *repDemande* pour savoir si le serveur vidéo est en train de répondre à une demande.
  - l'attribut *niveauQdsPossible* qui permet de savoir le niveau QdS que le serveur peut encore fournir alors qu'il répond déjà à une ou plusieurs demandes.
  - l'attribut *nbClient* qui permet de savoir à combien de clients le serveur répond afin de limiter le nombre de demandes traitées par celui-ci.

Présentation des méthodes :

- constructeur par défaut et constructeur avec paramètres (son nom, le niveau maximum QdS, le nombre de films, la liste des films).
- accesseurs et modifieurs des différents attributs.
- les différentes méthodes pour gérer la liste de films présentes sur le serveur (suppression et ajout de film, accesseur vers la liste des films ou vers un seul film de la liste).

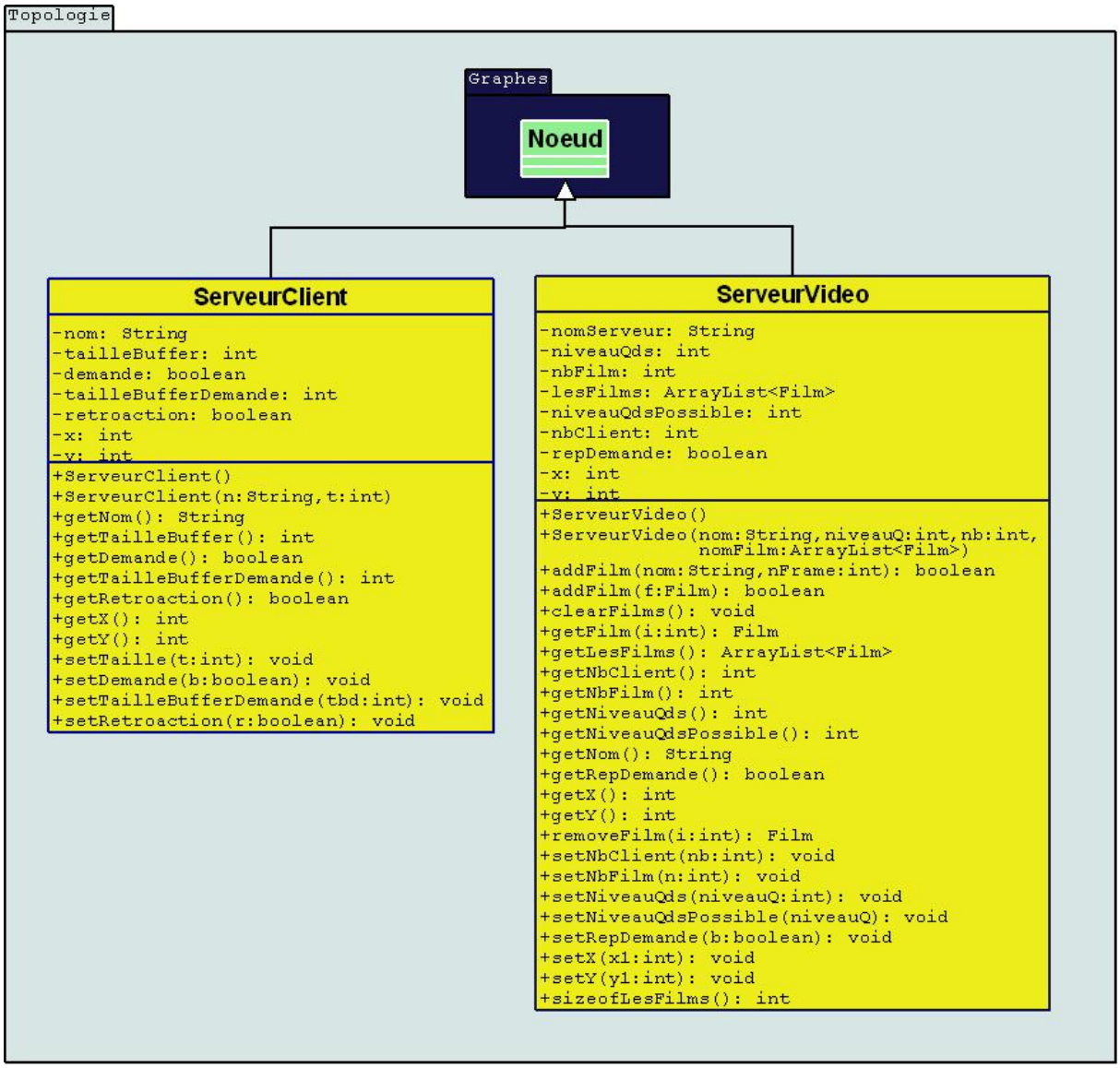


Figure 3. Schéma UML : Topologie 1

- ## Les différents flux :

Les différentes classes de flux servent principalement pour la représentation graphique des différents flux. Les objets créés héritent tous de la classe Arc du package Graphe.

Elles ont toutes sensiblement la même structure. Leurs attributs correspondent pour les flux entrants aux serveurs de départ du flux et pour les flux sortants aux serveurs d'arrivée du flux. A savoir que les objets de FluxEntrant et FluxSortant correspondent aux flux passant par le serveur maître et les objets de FluxEntrant2 et FluxSortant2 correspondent aux flux passant par le réseau internet.

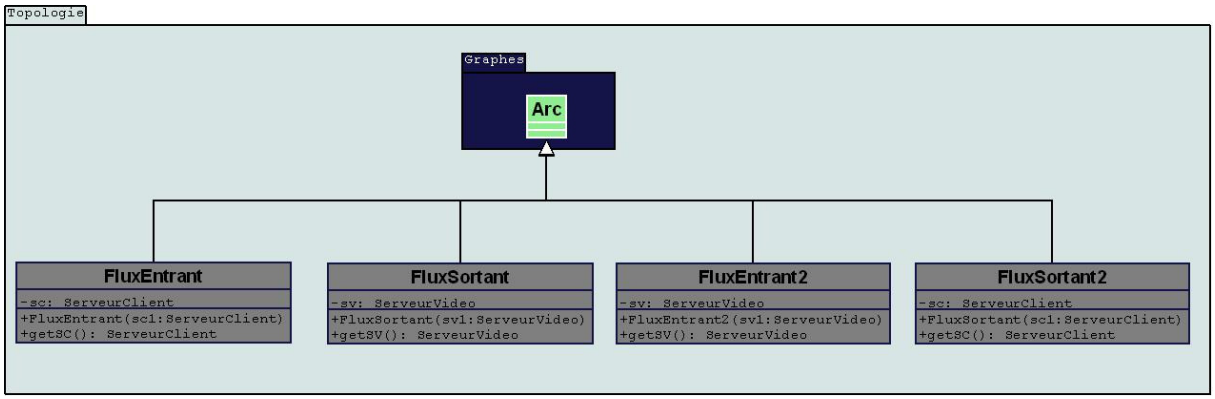


Figure 4. Schéma UML : Topologie 2

▪ Une Demande :

Présentation des attributs :

- attributs principaux :
  - *sc* : le serveur client qui fait la demande
  - *depart* : horaire de départ de la demande
  - *film* : le nom du film demandé
  - *nbFrames* : le nombre de frames de la demande
  - *qdsDemande* : la qualité de service demandée
  - *retroaction* : permet de savoir si la demande est une rétroaction

- autres attributs :
  - l'attribut *serveurTrouve* est l'indice du serveur vidéo trouvé pour répondre à la demande et l'attribut booléen *trouveServeurVideo* signale un serveur vidéo a été trouvé pour la demande.
  - l'attribut *depart2* correspond au départ d'une rétroaction.
  - l'attribut *serveurTrouveRetro* est l'indice du serveur vidéo trouvé pour répondre à une demande avant une rétroaction.
  - l'attribut booléen *reponse* pour savoir si le serveur vidéo a été mis en relation avec le serveur client.
  - l'attribut booléen *inactif* permet de savoir si la demande est active ou inactive.
  - l'attribut *departTransfert* permet de connaître l'horaire du départ du transfert entre serveur vidéo et serveur client.

Présentation des méthodes :

- constructeur avec paramètres (le serveur client, l'heure de départ, le nom du film demandé, le niveau QdS souhaité, le nombre de frames, la rétroaction).
- accesseurs et modifieurs des différents attributs.

Figure 5. Schéma UML : Topologie 3

- **Les autres classes :**

La classe *ServeurMaitre* est une classe qui sert juste à représenter physiquement le serveur maître mais on va se rendre compte plus tard de la véritable implémentation du maître.

La classe *Film* permet de créer un objet film qui contient le nom du film et son nombre de frames. Elle possède un constructeur avec paramètres (nom, nombre de frames) et des accesseurs et modifieurs vers les attributs *nom* et *nbFrame*. On se sert de ces objets dans les serveurs vidéo à travers une liste.

La classe *Reseau2* permet de représenter le réseau internet. L'attribut *nbDemande* permet de gérer la congestion de celui-ci. Elle comporte un constructeur par défaut ainsi qu'un accesseur et un modifieur vers son attribut.

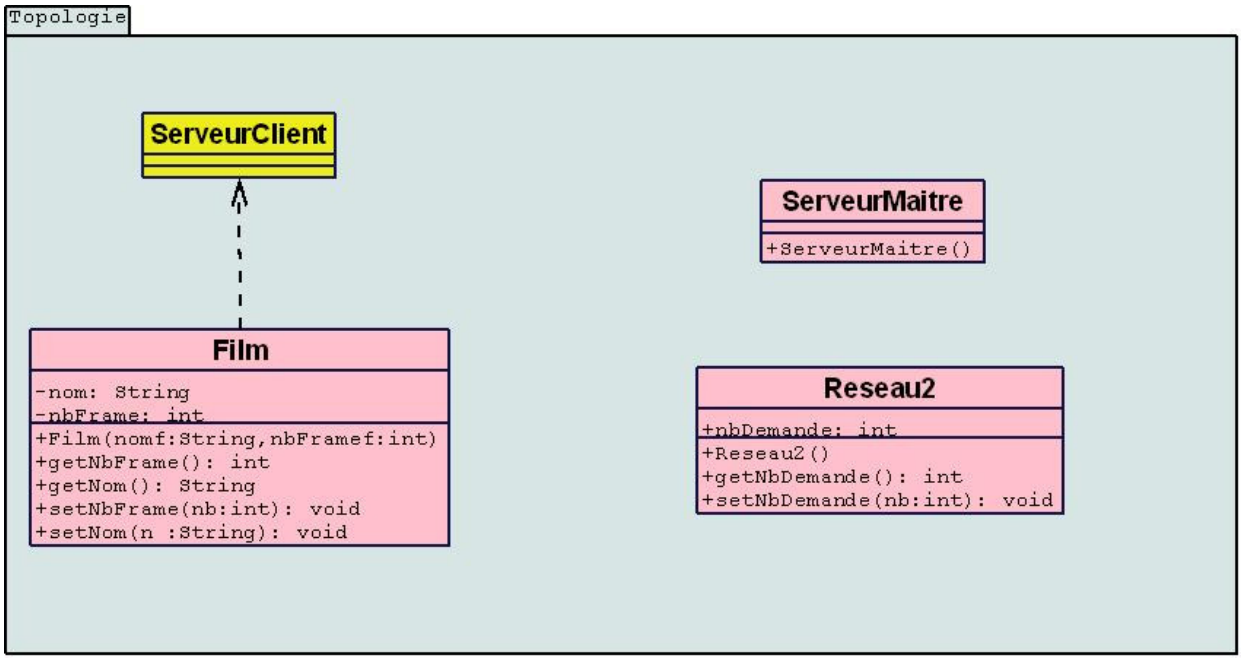


Figure 6. Schéma UML : Topologie 4

2.2.b. Package Transformation :

Le package transformation ne contient qu'une seule classe. La classe *Reseau* dans laquelle sont faites toutes les modifications sur le système multimédia. C'est pour cela que l'on peut considérer qu'il s'agit de l'implémentation du serveur maître.

Présentation des attributs :

- attributs principaux :
  - *graphe* : le graphe, avec les nœuds et les arcs, qui permet de donner au système une structure de graphe.
  - *lesDemandes* : la liste des demandes présentes dans le système
  - *sm* : la représentation physique du serveur maître
  - *res2* : la représentation du réseau internet
  - *nbDemande, nbServeurClient, nbServeurVideo, nbFluxEntrant, nbFluxSortant, nbFluxEntrant2 et nbFluxSortant2* : le nombre d'éléments de chaque objet présent dans le système.
  - *sauvegarde* qui permet de sauvegarder tous les changements qui se sont produits sur le réseau sous la forme d'un texte (nous l'expliciterons plus précisément avec les sauvegardes).

Présentation des méthodes :

- constructeur par défaut.
- accesseurs et modifieurs des différents attributs.
- méthodes d'ajout et de suppression des différents éléments constituant le système.
- les différentes méthodes permettant de gérer la liste de demandes (ajout, suppression, accesseurs, taille).
- une méthode *petitQds()* qui renvoie la plus petite QdS des serveurs vidéos pour permettre lors d'une demande que celle-ci est sur d'être traitée.

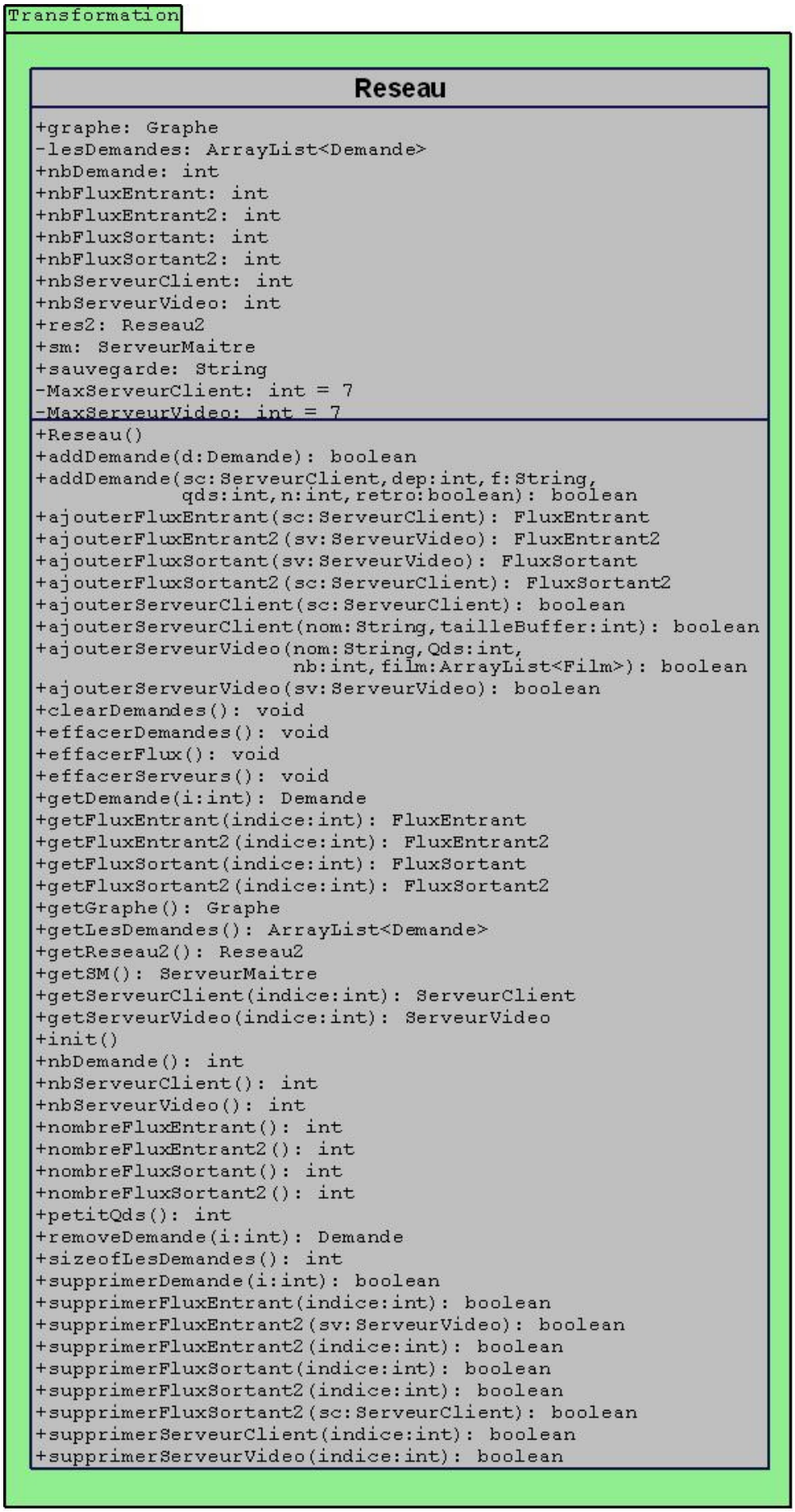


Figure 7. Schéma UML : Transformation

2.2.c. Package IHM :

Le package IHM comporte les différents composants de l'interface graphique. Les différentes classes présentes dans ce package permettent la gestion des événements effectués par l'utilisateur. Beaucoup de ces classes sont des fenêtres de dialogues et des fenêtres permettant à l'utilisateur d'interagir sur le réseau. Nous allons expliquer ces classes succinctement pour ensuite nous intéresser aux éléments importants dans l'implémentation du programme :

- *Simulation* : comporte l'exécutable du programme.
- *Interface* : est la fenêtre principale de l'interface graphique.
- *Onglet* : est le rassemblement des deux onglets graphes et description.
- *Graphe* : représente l'onglet graphe.
- *Description* représente l'onglet description.
- *DialogueSuppSV* : représente la fenêtre de dialogue permettant de supprimer un serveur vidéo.
- *DialogueSuppSC* : représente la fenêtre de dialogue permettant de supprimer un serveur client.
- *AjoutServeurVideo* : représente la fenêtre permettant d'ajouter un serveur vidéo.
- *AjoutServeurClient* : représente la fenêtre permettant d'ajouter un serveur client.
- *OServeurClient* : représente un onglet de serveur client dans lequel on peut modifier la taille de son buffer.
- *OngletServeurClient* : représente l'assemblage des différents *OServeurClient.*
- *FenServeurClient* : représente la fenêtre dans laquelle on introduit *OngletServeurClient.*
- *OServeurVideo* représente un onglet de serveur vidéo dans lequel on peut modifier son niveau QdS maximum, ajouter ou supprimer un film.
- *DialogueAjFilm* : représente la fenêtre de dialogue permettant d'ajouter un film.
- *DialogueSuppF* : représente la fenêtre de dialogue permettant De supprimer un film.
- *OngletServeurVideo* : représente l'assemblage des différents *OServeurVideo.*
- *FenServeurVideo* : représente la fenêtre dans laquelle on introduit *OngletServeurVideo.*
- *FenDemande* : représente la fenêtre permettant à l'utilisateur d'ajouter une demande.
- *FenDemandeSim* : représente la fenêtre permettant à l'utilisateur d'ajouter plusieurs demandes simultanées.
- *APropos* : petite fenêtre permettant de présenter succinctement le projet.

Remarque :

Un manuel d'utilisation est également à la disposition de l'utilisateur.

Passons à présent aux classes importantes de ce package :

- La classe *Partage* est la classe charnière entre l'interface graphique et le programme en lui-même. L'objet créé par cette classe permet le partage des données entre les méthodes.

- La classe *BarreMenu* est la classe permettant la création de la barre de menu avec ses différents items offrant les options à l'utilisateur. Mais cette classe comporte surtout les quatre méthodes importantes *open()*, *save()*, *open2()* et *save2()*.

open() et *save()* permettent à l'utilisateur d'ouvrir ou d'enregistrer un fichier pour pouvoir reprendre une simulation. La sauvegarde de fichier permet à l'utilisateur de sauvegarder la simulation à l'instant t de la sauvegarde. Lors de l'ouverture de ce fichier, la simulation reprendra à cet instant t. Cette sauvegarde se fait sous la forme d'un fichier texte dont l'extension est pour le moment initialiser à « .sim ».

Présentation du fichier de sauvegarde produit par *save():*

Les différentes données correspondent aux différents états d'un objet au moment de la sauvegarde (voir ANNEXE 1) :

Horaire au moment de la sauvegarde (heure, minute, seconde, temps courant en sec) :
```
0 0 34 34
```

Nombre de serveurs clients :
```
4
```
Pour chaque serveur client :
- nom, taille du buffer, buffer consommé durant une demande, 1 si demande, 1 si rétroaction .

```
ServeurClient1 120 0 1 1
```

Nombre de serveurs vidéos :
```
4
```
Pour chaque serveur vidéo :
- nom, QdS max, nombre de films, QdS possible pendant une demande, 1 si le maître communique avec lui.
- liste des films : nom du film, nombre de frames du film

```
ServeurVideo4 150 3 50 0
Film1 360
Film4 360
Film5 340
```

Nombre de demandes:
```
4
```
Pour chaque demande:
- nom du demandeur, heure de départ, film demandé, QdS demandée, nombre de Frames, 1 si rétroaction
- 0 si la demande est active.
- 1 si un serveur a été trouvé pour répondre à la demande.
  → index du serveur trouvé si la valeur est à 1
- 1 si un serveur vidéo a répondu à la demande
  → départ du transfert si la valeur est 1

```
ServeurClient3 26 Film5 100 340 0
0
1
3
1
32
```

Texte du panneau des événements de l'onglet description :
```
00 : 00 : 26: Le serveur Client "ServeurClient1" a demandé le film : Film5.
```

Texte de la sauvegarde (précéder par sauv):

```
sauv
tps 26
dem
0 26 Film5 15
…
```

open2() et *save2()* permettent à l'utilisateur d'ouvrir ou d'enregistrer un fichier pour pouvoir visualiser toute une simulation. Toute au long d'une simulation, une sauvegarde est faite sous forme de texte. Lors de la sauvegarde, ce texte est introduit dans notre fichier. Lors de l'ouverture de ce fichier, l'utilisateur pourra visualiser la simulation depuis son début. Cette sauvegarde se fait sous la forme d'un fichier texte dont l'extension est pour le moment initialiser à « .vis ».

Présentation du fichier de sauvegarde produit par *save2():*

Les différentes données correspondent aux différentes transformations que le réseau a connues (voir ANNEXE 2) :

Horaire de la transformation précédée de tps pour savoir qu'il s'agit d'une nouvelle transformation:

```
tps 0
```

Un identfiant pour chaque transformation :

- sc pour l'ajout d'un serveur client puis :
  → nom, taille buffer
- sv pour l'ajout d'un serveur vidéo puis :
  → nom, QdS max, nombre de film
  →pour chaque film : nom, nombre de frames
- dem pour l'ajout d'une demande puis :
  →indice du demandeur, heure de départ de la demande, nom du film demandé, QdS demandée
- AjF pour l'ajout d'un film puis :
  →indice du serveur vidéo sur lequel le film a été ajouté, nom du film, nombre de frames
- SuppF pour la suppression d'un film puis :
  →indice du serveur vidéo sur lequel le film a été supprimé, indice du film
- supSC pour la suppression d'un serveur client puis :
  →indice du serveur client à supprimer
- supSV pour la suppression d'un serveur vidéo puis :
  →indice du serveur vidéo à supprimer
- modTB pour la modification de la taille du buffer d'un serveur client :
  →indice du serveur client, nouvelle taille du buffer
- modQDS pour la modification du QdS max d'un serveur vidéo :
  →indice du serveur vidéo, nouveau QdS max

Horaire de la sauvegarde précédé de fin pour savoir qu'il s'agit de l'horaire de sauvegarde:

```
fin
16
```

- La classe *BarreBouton* est l'une des classes centrales de ce programme. En effet, celle-ci contient le timer ainsi que son écouteur qui va permettre de traiter une visualisation par la méthode *traiteVisu(chemin, sec)* et gérer l'évolution des demandes dans le temps par l'intermédiaire d'une méthode *traiteDemande(sec)*. La méthode *traiteVisu(chemin, sec)* est une méthode d'ouverture du fichier .vis que l'on souhaite visualiser. A chaque seconde, on accède au fichier dont le chemin a été passé en paramètre de

la fonction et dès que l'horaire de la transformation correspond à la seconde courante (voir ci-dessus), on traite cette transformation.

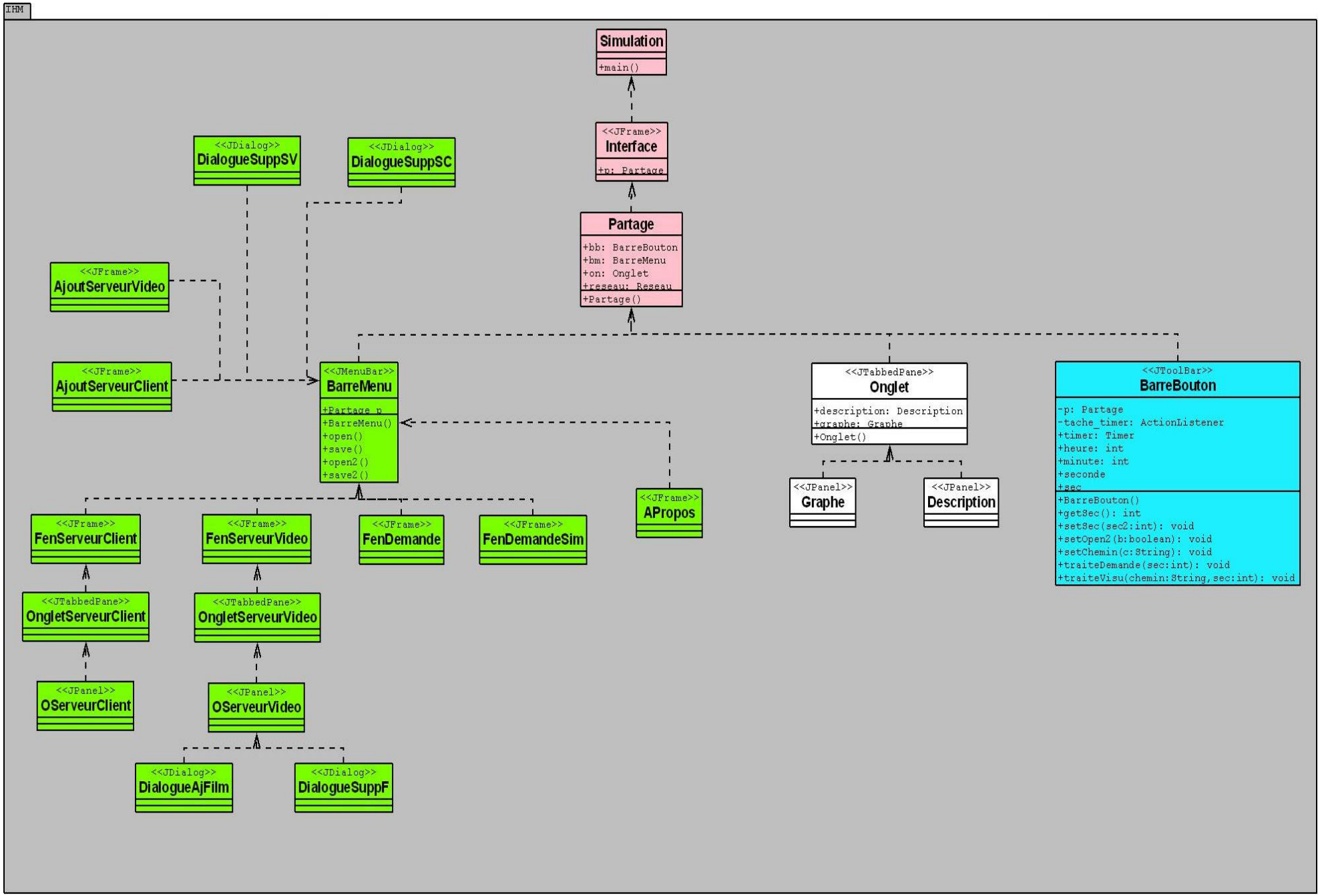
La fonction *traiteDemande(sec)* est la méthode centrale de la simulation. En effet, celle-ci permet à chaque seconde de traiter les demandes effectuées sur le réseau. Etant la fonction répondant à la problématique du projet, nous allons étudier son algorithme de plus prêt.

Pseudo-code de l'algorithme de la méthode *traiteDemande(sec)* :

```
Données : sec=seconde courante

POUR i allant de 0 à nombre de demandes
   SI la demande est active
      SI sec<départ de la demande+2
         On est au début de la demande
         SI la demande est une rétroaction
            Le client effectue une rétroaction au maître
         SINON c'est une demande normale au maître
      SINON
         Le client n'effectue plus sa demande au maître
         SI le serveur maître n'a pas encore trouvé de serveur
            SI la demande est une rétroaction && sec=départ de la rétroaction+2
               i=indice du serveur vidéo qui répondait à l'ancienne demande.
            SINON c'est une demande normale
               SI on a pas encore trouvé de serveur vidéo pour répondre à la
                   demande
                  i=sec-(départ de la demande+2)
                  i est l'indice du serveur répondant au serveur maître
                  SI il est possible pour le serveur de fournir la QdS
                     demandée et qu'il ne s'occupe pas de plus de 2 serveurs
                     POUR chaque film du serveur
                        SI on trouve le film demandé
                           On a trouvé le serveur vidéo
                        SINON ce n'est pas le bon serveur
               SI il s'agissait du dernier serveur à visiter
                  SI c'est une rétroaction
                      on reprend le chargement avec l'ancien serveur vidéo
                      avec la plus grosse QdS qu'il peut fournir.
                  SINON
                     départ de la demande=sec-1 ;
      SINON
         Le serveur vidéo trouvé est mis en relation avec le serveur client.
         départ transfert=sec.
      SINON SI sec<départ tranfert+3
                  Pendant 2 secondes, seul le buffer du serveur client est
                  rempli avec la QdS demandée - congestion du réseau (simulée
                  avec le nombre de demandes sur le réseau)
            SINON
                 Remplissage du buffer avec QdS demandée-congestion du réseau
                 Vidage du buffer de 25 frames
                 SI le nombre de frames<0
                      Terminaison avec succès
                      La demande devient inactive
                 SI il reste moins de 25 frames dans le buffer
                      Rétroaction du client
```

Figure 8. Schéma UML : Ihm



2.2.d. Package Graphe :

Le package graphe n'est que peu utilisé pour le moment dans la simulation. Il permet juste au système d'avoir une structure de graphe, pour peut-être dans l'avenir améliorer le programme. Il est constitué des classes communes dans la représentation de graphes, une classe *Nœud* représentant les nœuds du graphe, une classe *Arc* représentant les arcs du graphe et une classe *Graphe* constituée de listes de ces derniers.

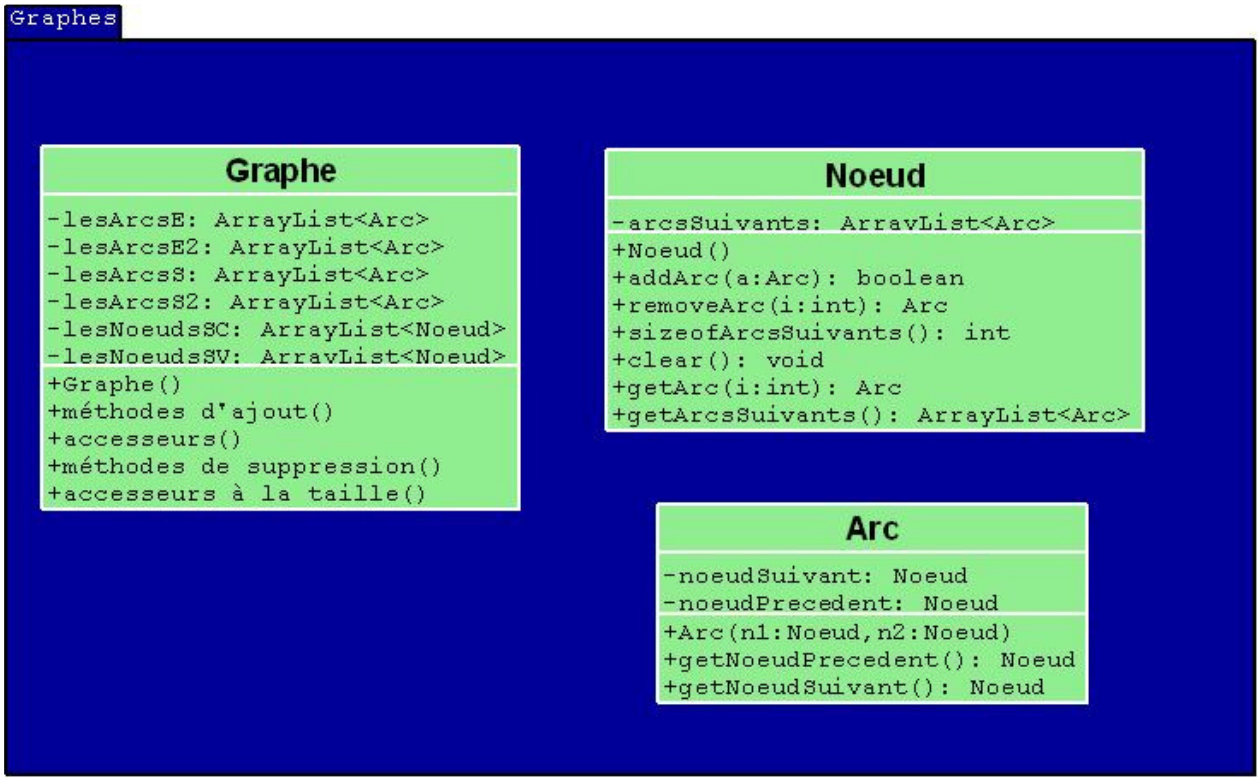


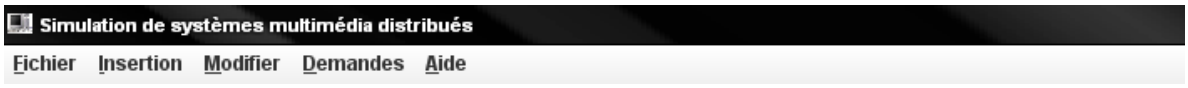
Figure 9. Schéma UML : Graphe

CHAPITRE 3 :

Manuel d'utilisation

3.1. La barre des menus

La barre des menus permet d'accéder aux principales options du programme :



3.1.a. Le menu Fichier :

Menu Fichier

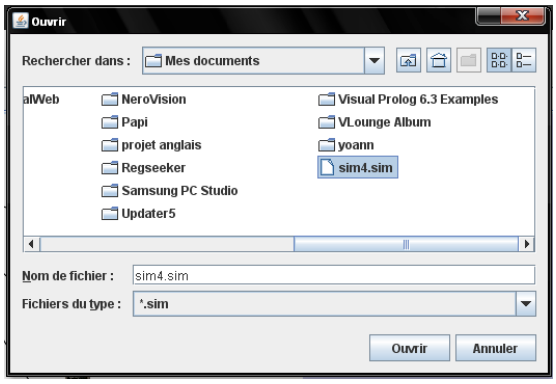
Le menu Fichier donne les fonctionnalités généralement applicables à tous les fichiers : nouveau fichier, ouverture, sauvegarde et fermeture de fichier.

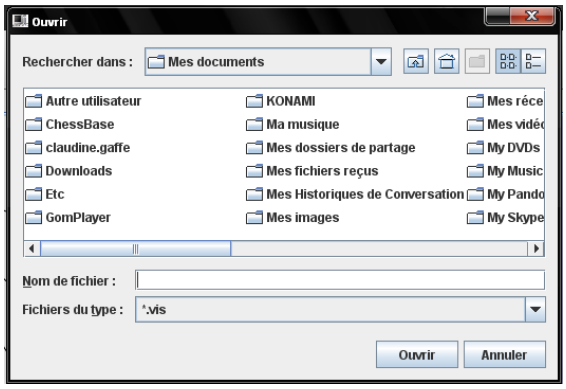
- Item "Nouvelle Simulation" :

Cet item permet la création d'une nouvelle simulation. Toutes les données sont alors réinitialisées. C'est pour cela que lors de son appel, une fenêtre de dialogue propose à l'utilisateur de sauvegarder les données de la simulation en cours.

- Item "Ouvrir" :

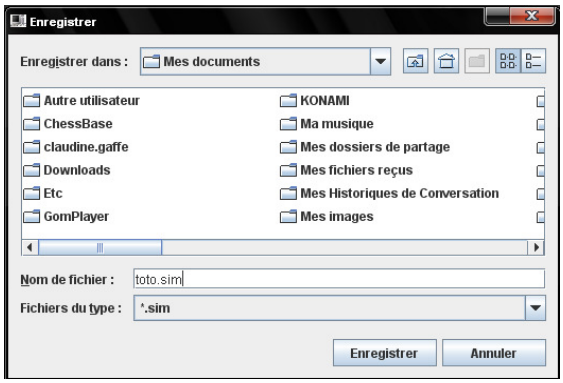
Cet item propose à l'utilisateur soit d'ouvrir une simulation, soit d'ouvrir une visualisation, c'est-à-dire que l'utilisateur peut reprendre une simulation à partir d'un instant t ou visualiser la simulation puis la reprendre à l'instant ou celle-ci fut enregistré.

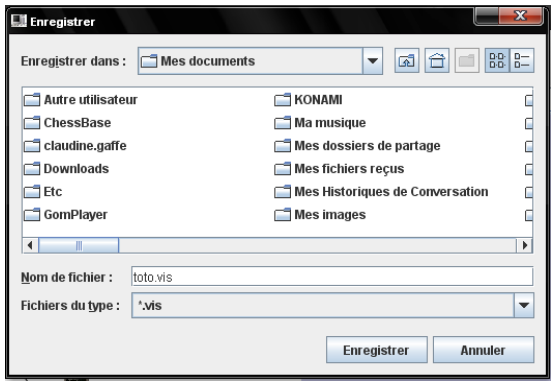




- Item “Enregistrer” :

L’item Enregistrer ouvre une fenêtre permettant à l’utilisateur de sauvegarder une simulation en cours ou depuis le début, c’est-à- dire que l’utilisateur pourra reprendre la simulation à l’instant de la sauvegarde ou reprendre sa simulation depuis le début. Le fichier de sauvegarde doit contenir l’extension .sim pour la sauvegarde d’une simulation en cours et .vis pour la sauvegarde d’une visualisation comme dans les exemples ci-dessous.



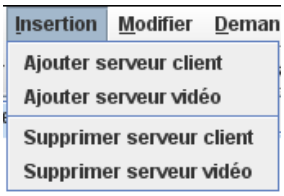


- Item “Quitter” :

L’item Quitter permet de quitter le programme proprement. Comme lors d’une nouvelle simulation, une fenêtre de dialogue propose de sauvegarder les données de la simulation en cours.

3.1.b. Le menu insertion :

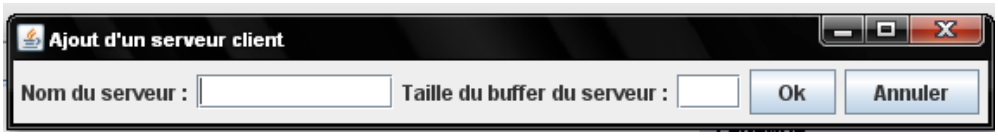
Le menu insertion permet l’ajout et la suppression de serveurs clients et de serveurs maîtres.



Menu Insertion

- Item “Ajouter serveur client” :

Cet item permet l’ouverture d’une fenêtre permettant d’ajouter un nouveau serveur client à la simulation.



L’utilisateur doit entrer le nom du serveur client sachant qu’il ne peut contenir d’espaces et la taille de son buffer qui doit être un nombre sinon des messages d’erreurs s’affichent à l’écran.

- Item “Ajouter serveur vidéo” :

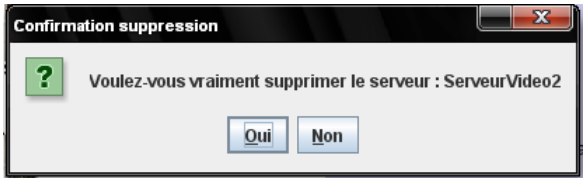
Cet item permet l’ouverture d’une fenêtre d’ajout de nouveau serveur vidéo au système multimédia.

L’ajout d’un serveur vidéo se fait en deux étapes. L’utilisateur doit tout d’abord donner le nombre de films présents sur le serveur, puis cliquer sur le bouton « Ajouter films…». Les champs de saisie correspondant aux noms des films et au nombre de frames de chaque film s’affichent alors.

L’utilisateur peut alors valider les données qu’il a entrées. Tout comme pour l’ajout d’un serveur client, si tous les champs ne sont pas remplis ou si des données sont considérées comme inexactes des messages d’erreurs s’affichent à l’écran.

- Item “Supprimer serveur client” et “Supprimer serveur vidéo” :

L’item “Supprimer serveur client (resp.vidéo)” permet l’ouverture d’une fenêtre donnant le choix à l’utilisateur de supprimer un serveur client (resp. vidéo) parmi la liste des serveurs clients (resp. vidéos) du réseau.



Lorsque l’utilisateur valide la suppression d’un serveur, une fenêtre de confirmation s’affiche pour être sur du choix de celui-ci.

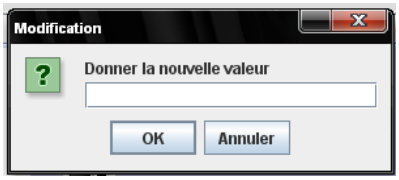
3.1.c. Le menu Modifier :

Le menu modifier permet la modification des données sur les serveurs clients et sur les serveurs vidéos.

Menu Modifier

- Item "Serveurs clients…" :

Cet item permet à l'utilisateur de modifier la taille du buffer d'un serveur client. Une fenêtre s'affiche avec un onglet pour chaque serveur client. Pour modifier la taille du buffer, l'utilisateur doit cliquer sur le bouton « Modifier » ; une fenêtre de dialogue s'affiche pour que l'utilisateur puisse entrer une nouvelle valeur.



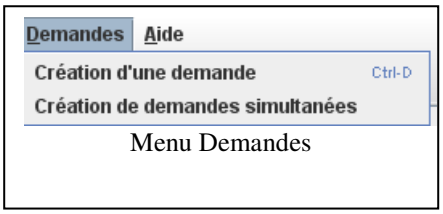
- Item "Serveurs videos" :

Tout comme l'item précédent, cet item permet d'ouvrir une fenêtre dans laquelle chaque serveur vidéo est présenté sous forme d'onglet.

L'utilisateur peut alors soit modifier le maximum de qualité de service du serveur vidéo, soit ajouter un film en cliquant sur le bouton « Ajouter film », soit supprimer un film en cliquant sur le bouton « Supprimer film». Pour l'ajout, une fenêtre s'affiche afin que l'utilisateur saisisse les données du film qu'il veut ajouter au serveur vidéo. Pour la suppression, l'utilisateur choisit le film qu'il veut supprimer. Lors de la suppression, une fenêtre de confirmation s'affiche pour être sur du choix de l'utilisateur.

3.1.d. Le menu Demandes :

Le menu Demandes permet la création soit d'une seule demande, soit de demandes simultanées. Lorsqu' une demande est créée, le menu Insertion devient inactif pour éviter de rencontrer des problèmes lors du traitement de celle-ci.



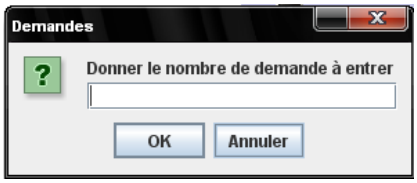
Menu Demandes

- Item "Création d'une demande" :

Cet item permet l'ouverture d'une fenêtre permettant à l'utilisateur de créer une demande. L'utilisateur choisit le serveur qui fait la demande, le film demandé et saisit la qualité de service demandée. Si cette dernière est plus grande que la plus petite valeur des QdS des serveurs vidéos, un message d'erreur s'affiche car il est possible que celle-ci ne soit pas traitée.

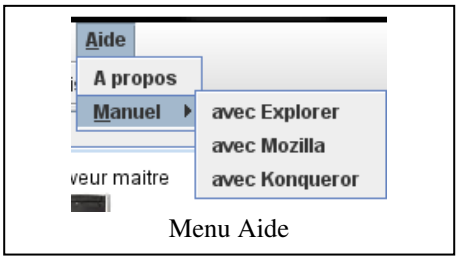
- Item "Création de demandes simultanées" :

La création de demandes simultanées s'effectue en deux étapes. Tout d'abord, il est demandé à l'utilisateur le nombre de demandes simultanées qu'il veut créer. Si le nombre de demandes est supérieur au nombre de serveurs clients, une fenêtre d'erreur s'affiche car il est impossible d'effectuer des demandes simultanées du même serveur client.



Ensuite comme précédemment, l'utilisateur choisit le serveur client qui effectue la demande, le film demandé et saisit la qualité de service demandée. Si deux demandes sont effectuées du même serveur, un message d'erreur s'affiche alors à l'écran.

3.1.e. Le menu Aide :



Menu Aide

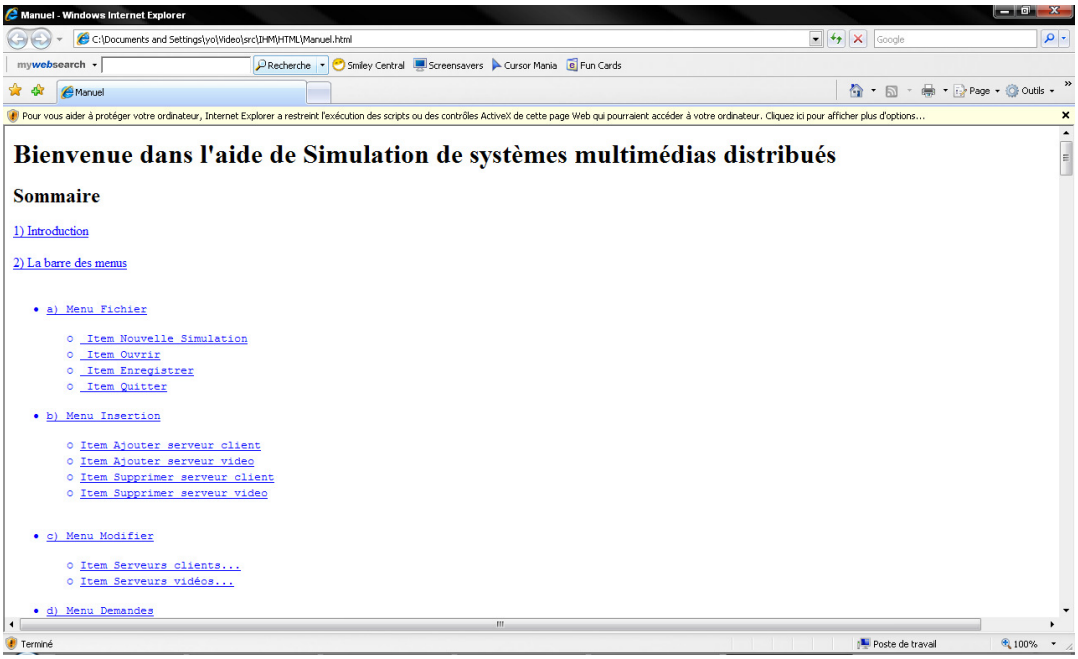
Le menu Aide offre les différentes fonctionnalités d'une aide.

- Item "A propos" :

Cet item propose un A propos présentant le projet (sujet, tuteur, …).

- Item "Manuel" :

Cet item propose à l'utilisateur d'ouvrir un manuel dans différents navigateurs. Si un problème se produit durant l'ouverture du manuel, un message indique à l'utilisateur le répertoire dans lequel se trouve celui-ci.



3.2. La barre des boutons

La barre de boutons est composée d'un affichage de l'heure courante de la simulation et de deux boutons pour interagir sur cette simulation :

- Lancer/Stop : lance et met en pause la simulation. Lorsque la simulation est mise en pause, le menu Demandes devient inactif.
- Réinitialiser : remet le réseau dans son état initial afin d'effectuer une nouvelle simulation. Une proposition de sauvegarde est faite alors avant de réinitialiser les données.

3.3. Les onglets

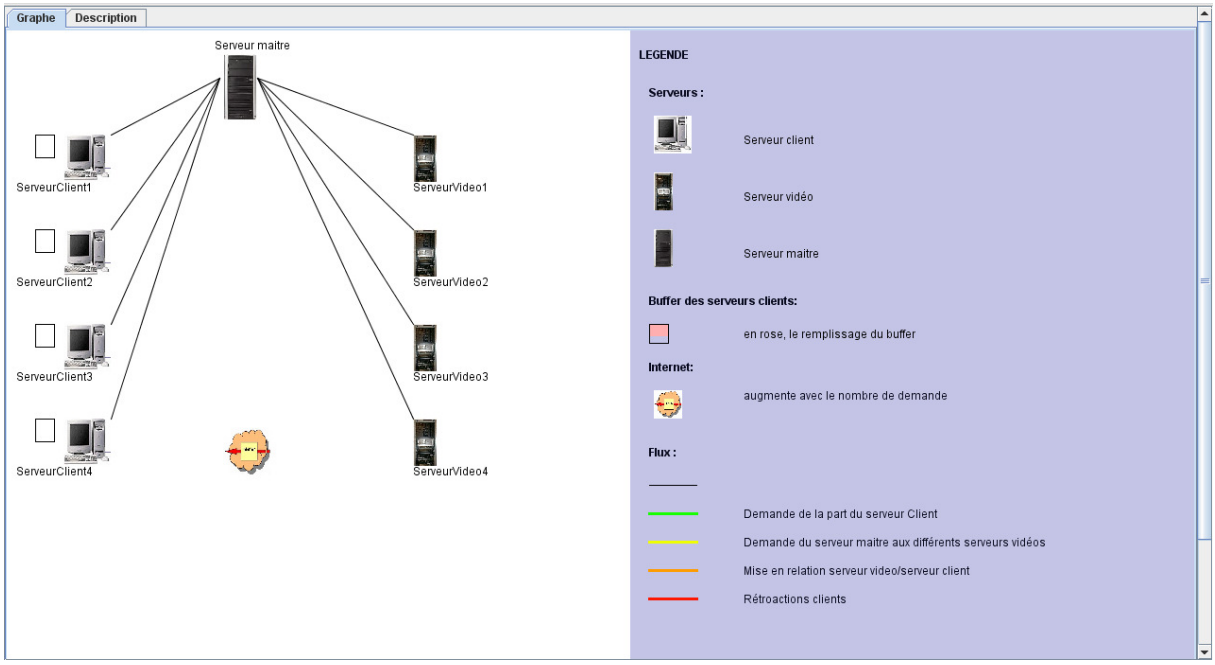
3.3.a. L'onglet Graphe

Il donne une représentation graphique du réseau avec les différents serveurs (maître, clients et vidéos) et une représentation du réseau internet avec sa congestion.

Les serveurs clients sont représentés par des ordinateurs avec à leur côté une représentation de leur buffer par un rectangle qui se remplit de rose lors du transfert des données du serveur vidéo vers ceux-ci et les serveurs vidéos sont représentés par des tours d'ordinateur. Ils sont bien évidemment reliés par des arcs au serveur maître. Le réseau internet est représenté par un nuage qui augmente de taille avec la congestion.

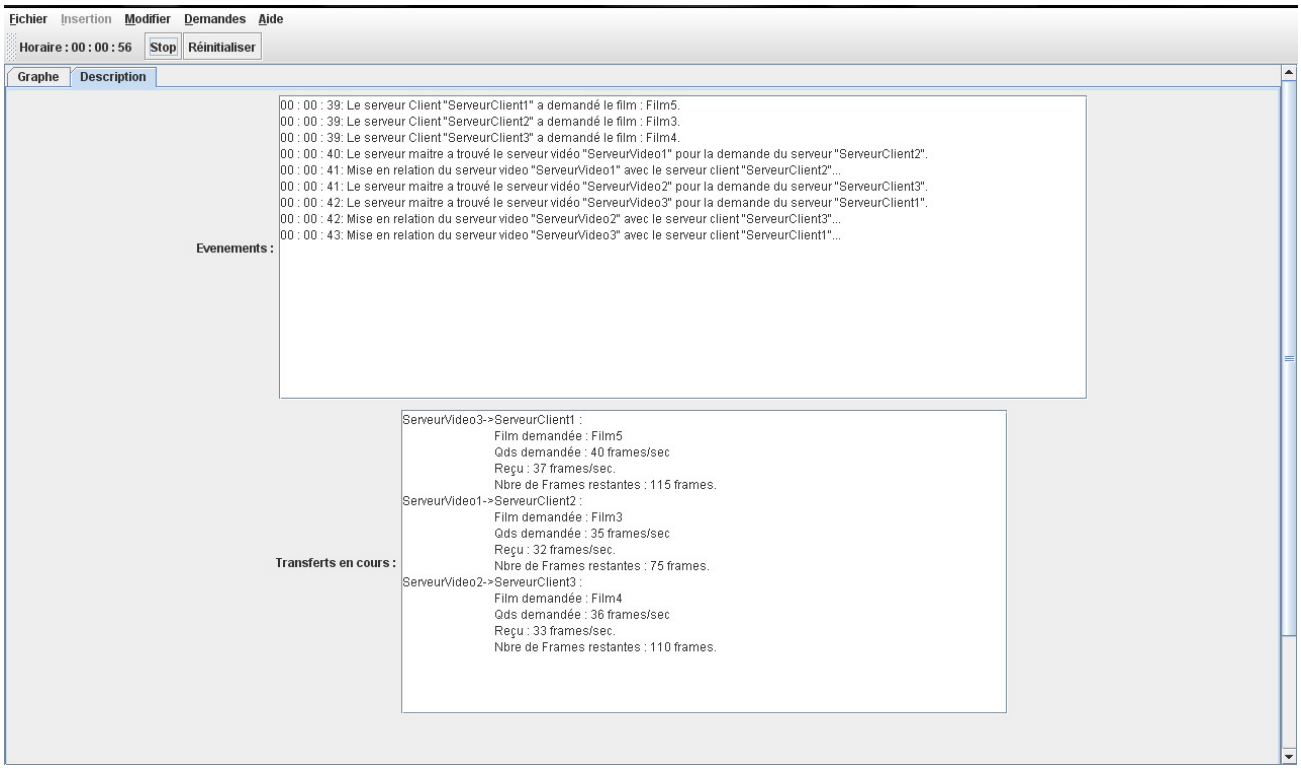
Une légende est présente aussi pour permettre à l'utilisateur de comprendre les différentes étapes qui s'y déroulent.

Il est possible pour l'utilisateur de bien visualiser l'évolution des demandes grâce aux changements de couleur des arcs selon les états de la demande.



3.3.b. L'onglet Description :

L'onglet Description permet une visualisation écrite des événements qui se déroule dans le système. Il est divisé en deux panneaux : un présentant les événements principaux (demande au maître, trouve le serveur, mise en relation, rétroaction…) et l'autre présentant les transferts de données en cours en donnant le nom des deux serveurs mis en relation, le film demandé, la QdS demandée , ce qui est reçu et le nombre de frames restants à être visionner.


Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 00 : 56
Stop
Réinitialiser
Graphe
Description
Evenements :
00 : 00 : 39: Le serveur Client "ServeurClient1" a demandé le film : Film5.
00 : 00 : 39: Le serveur Client "ServeurClient2" a demandé le film : Film3.
00 : 00 : 39: Le serveur Client "ServeurClient3" a demandé le film : Film4.
00 : 00 : 40: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1" pour la demande du serveur "ServeurClient2".
00 : 00 : 41: Mise en relation du serveur video "ServeurVideo1" avec le serveur client "ServeurClient2"...
00 : 00 : 41: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo2" pour la demande du serveur "ServeurClient3".
00 : 00 : 42: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient1".
00 : 00 : 42: Mise en relation du serveur video "ServeurVideo2" avec le serveur client "ServeurClient3"...
00 : 00 : 43: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient1"...
Transferts en cours :
ServeurVideo3->ServeurClient1 :
Film demandée : Film5
Qds demandée : 40 frames/sec
Reçu : 37 frames/sec.
Nbre de Frames restantes : 115 frames.
ServeurVideo1->ServeurClient2 :
Film demandée : Film3
Qds demandée : 35 frames/sec
Reçu : 32 frames/sec.
Nbre de Frames restantes : 75 frames.
ServeurVideo2->ServeurClient3 :
Film demandée : Film4
Qds demandée : 36 frames/sec
Reçu : 33 frames/sec.
Nbre de Frames restantes : 110 frames.

CHAPITRE 4 :
Conclusion

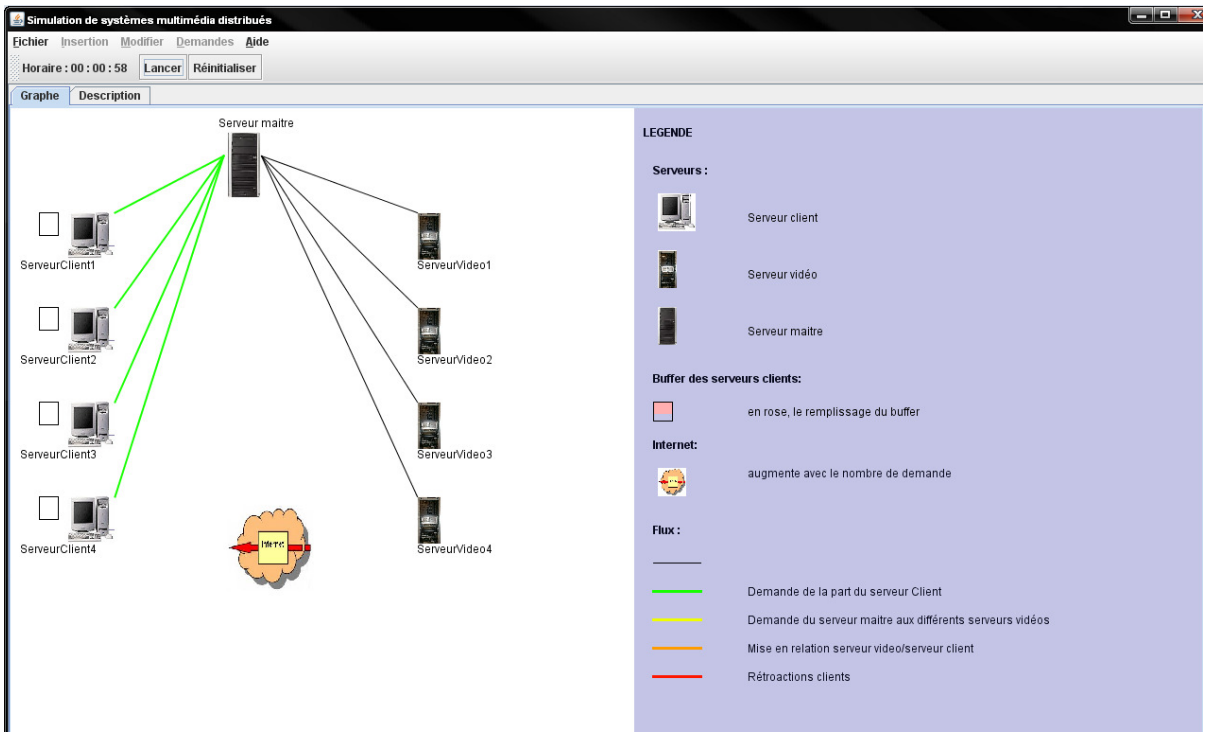
4.1. Résultats

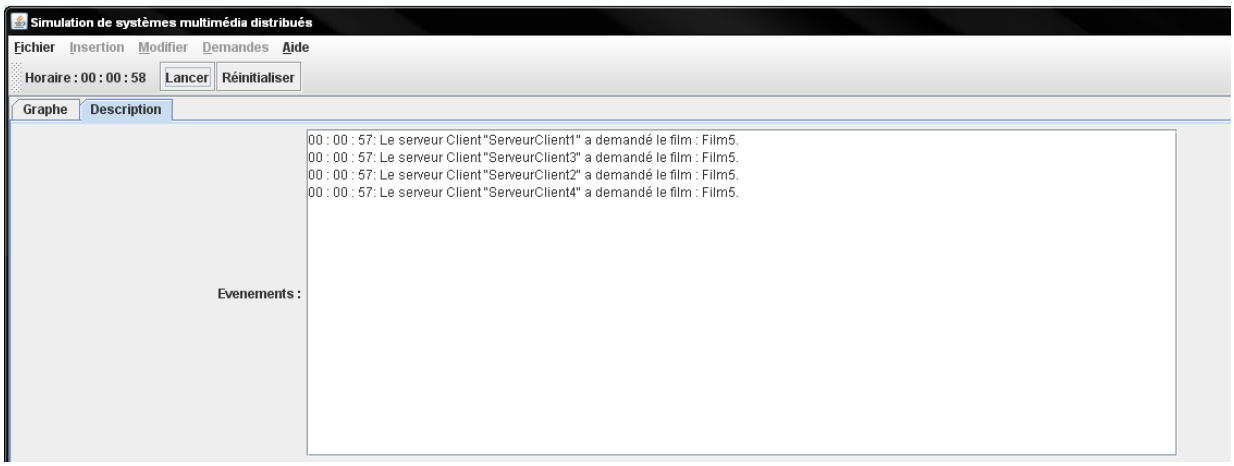
Les résultats paraissent assez convaincants bien que de nombreuses améliorations peuvent surement être apportées au programme. Les principales exceptions semblent être gérées et les différentes étapes d'une demande sont visualisables. En effet, nous allons pouvoir essayer de visualiser le fonctionnement du programme à travers divers exemples (avec des serveurs vidéos de 120, 160, 130 et 150 de QdS maximum).

Exemple1 : On effectue quatre demandes simultanées d'un même film avec un fort niveau QdS demandé par chaque serveur client.

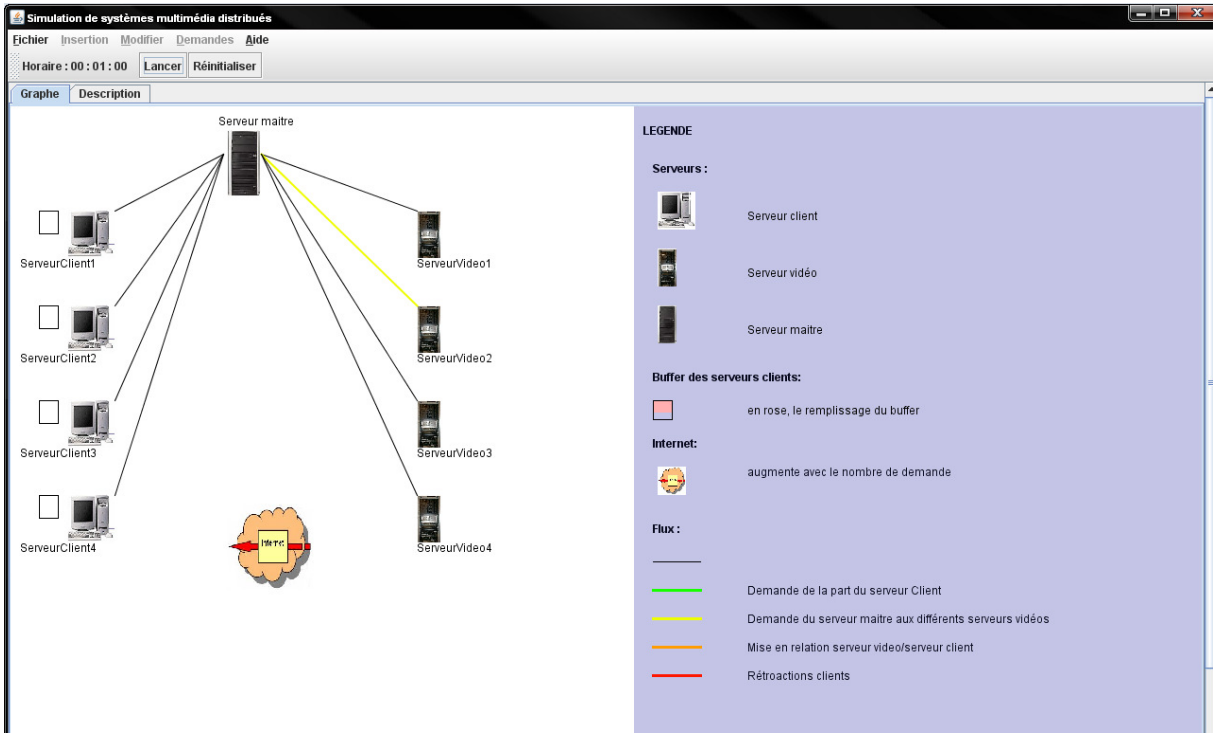
Création de demandes simultanées
Serveur demandeur ServeurClient1 Film demandé : Film5 Qualite de service demandée : 70 frame/sec.
Serveur demandeur ServeurClient3 Film demandé : Film5 Qualite de service demandée : 35 frame/sec.
Serveur demandeur ServeurClient2 Film demandé : Film5 Qualite de service demandée : 90 frame/sec.
Serveur demandeur ServeurClient4 Film demandé : Film5 Qualite de service demandée : 65 frame/sec.
Ok
Annuler

1ère étape : Les serveurs envoient leur demande au serveur client.

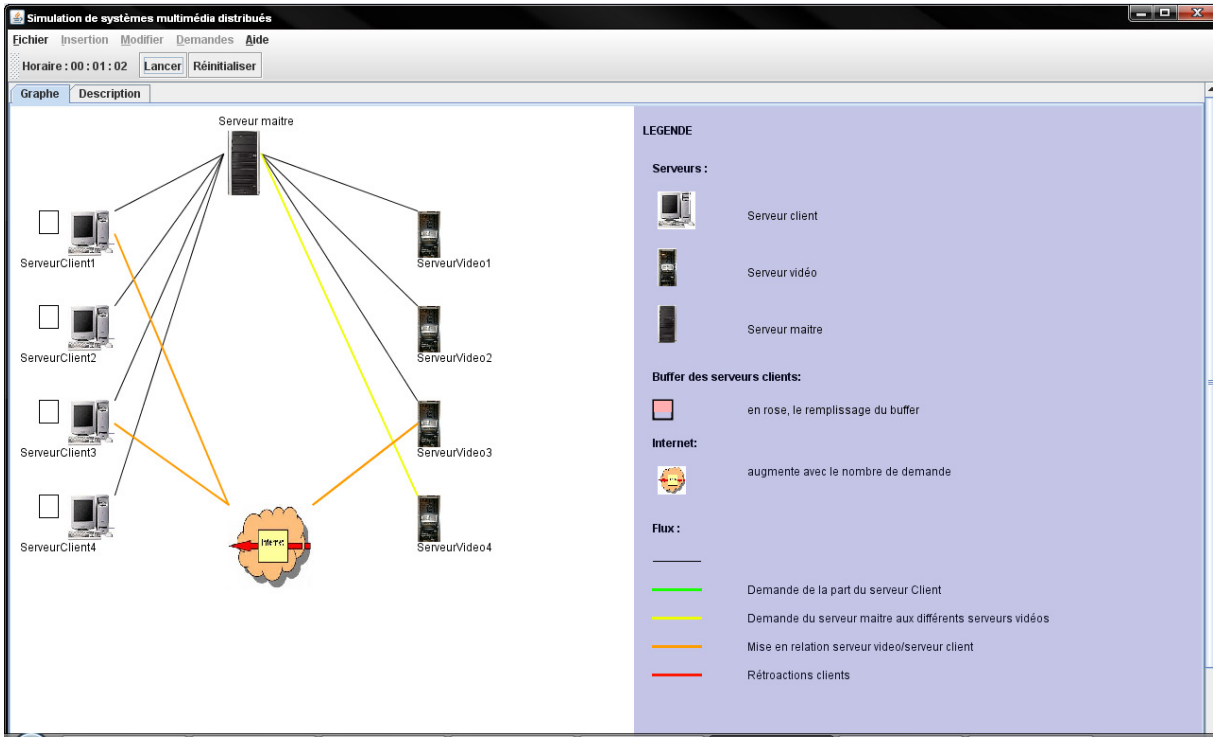


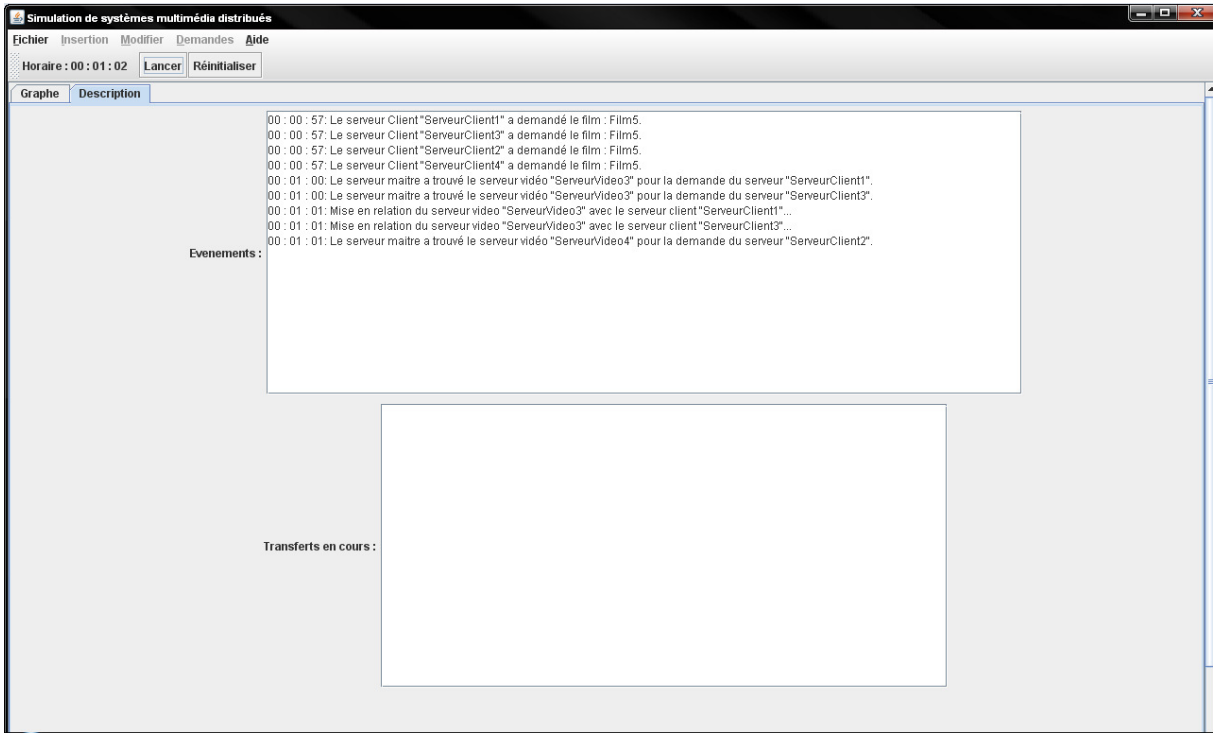

Simulation de systèmes multimédia distribués
Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 00 : 58
Lancer
Réinitialiser
Graphe
Description
00 : 00 : 57: Le serveur Client "ServeurClient1" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient3" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient2" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient4" a demandé le film : Film5.
Evenements :

2ème étape : Le serveur maître recherche pour chaque demande le serveur vidéo pouvant répondre.

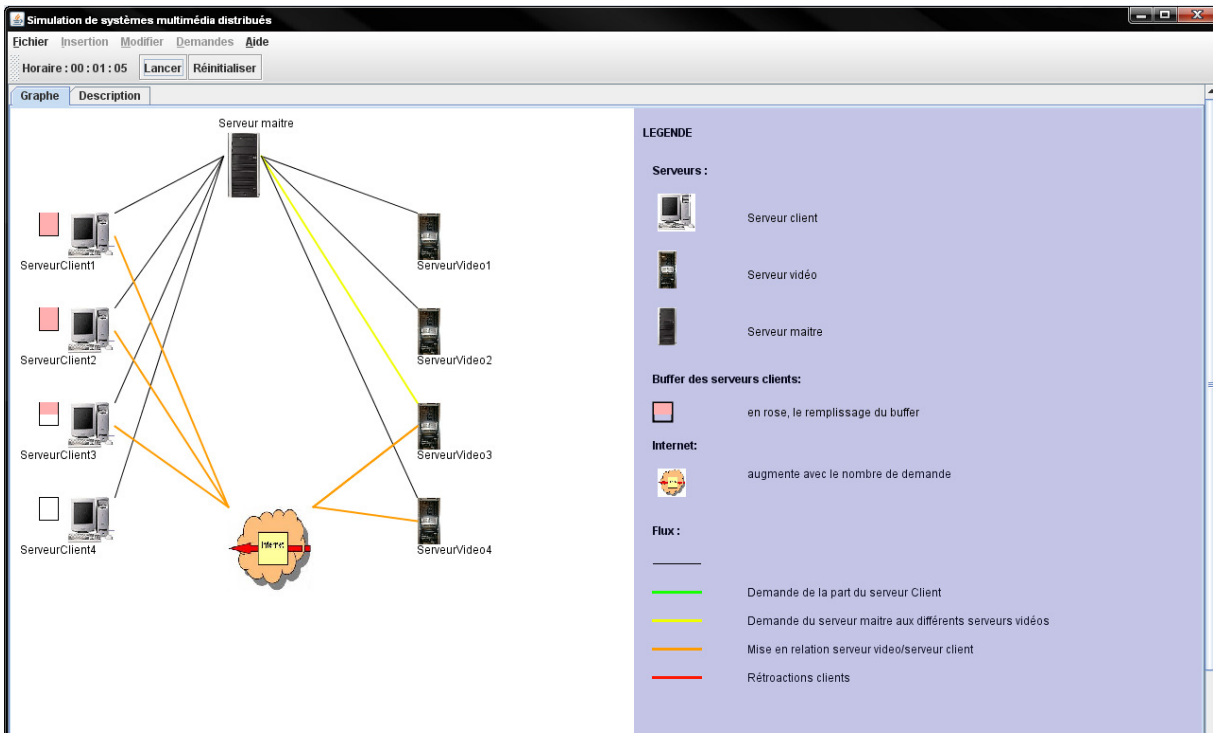


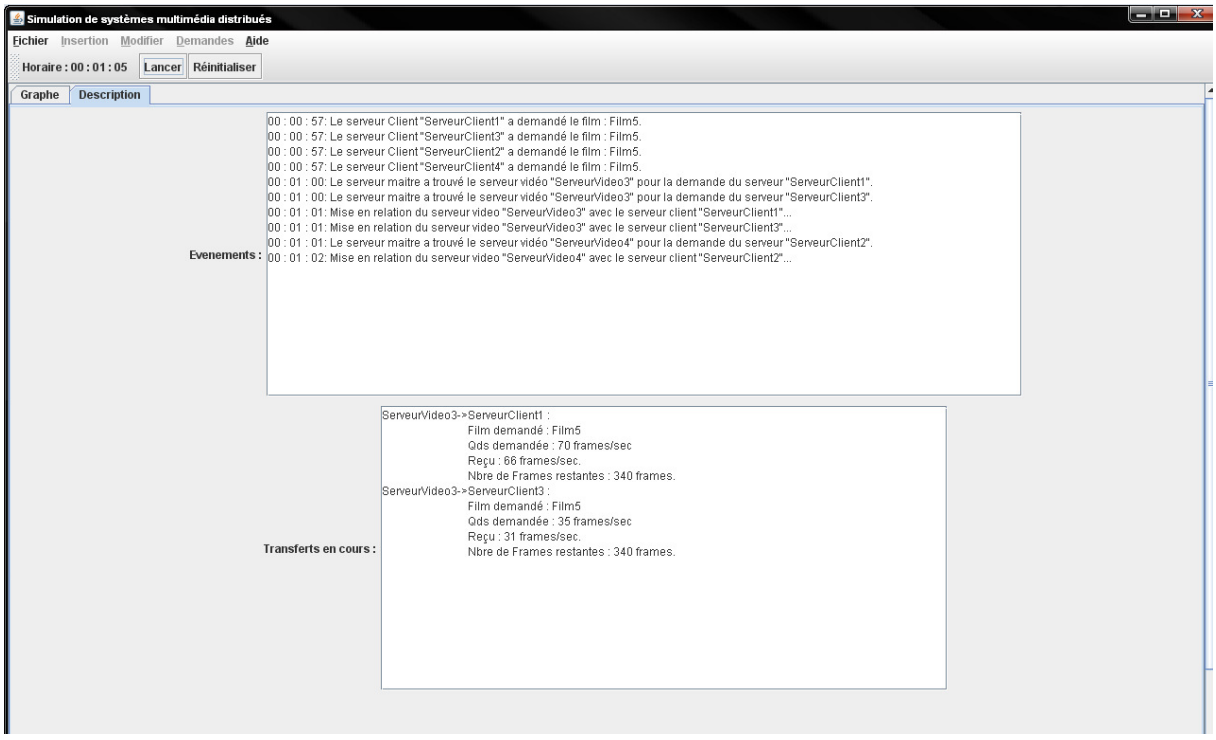
3ème étape : Le maître a trouvé le serveur 3 pour répondre à la demande des clients 1 et 3 mais ayant déjà deux demandes en cours de traitement, il ne peut répondre à la demande des deux autres clients.



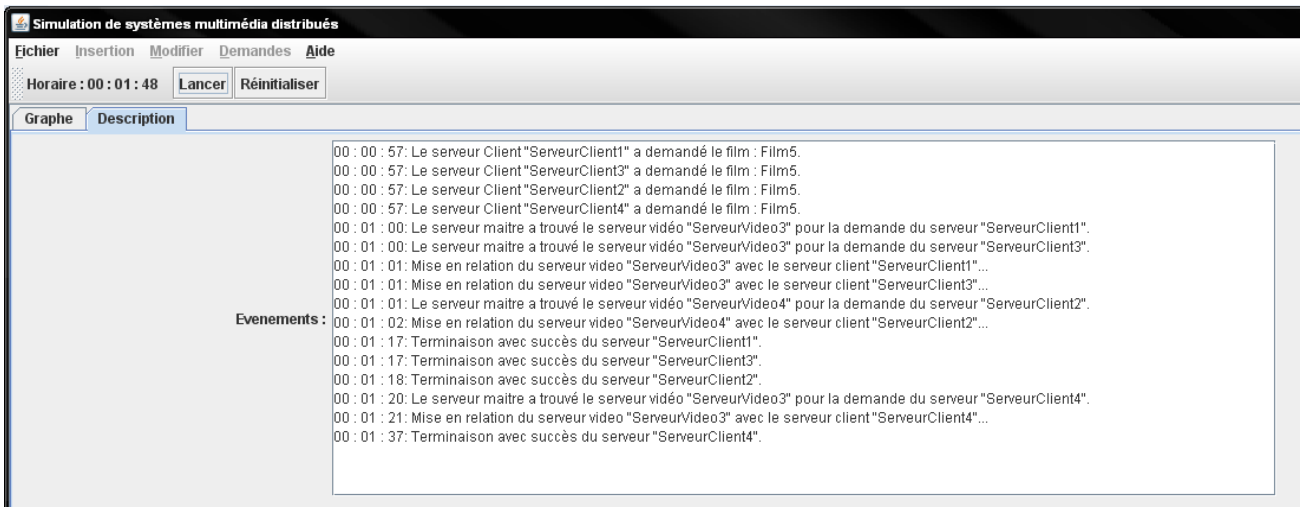

Simulation de systèmes multimédia distribués
Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 01 : 02
Lancer
Réinitialiser
Graphe
Description
Evenements :
00 : 00 : 57: Le serveur Client "ServeurClient1" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient3" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient2" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient4" a demandé le film : Film5.
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient1".
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient3".
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient1"...
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient3"...
00 : 01 : 01: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo4" pour la demande du serveur "ServeurClient2".
Transferts en cours :

4ème étape : Les serveurs clients commencent à recevoir leur film. Remarquons que le serveur 4 n'a toujours pas eu de réponse les serveurs vidéo étant surchargés.




Simulation de systèmes multimédia distribués
Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 01 : 05
Lancer
Réinitialiser
Graphe
Description
Evenements :
00 : 00 : 57: Le serveur Client "ServeurClient1" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient3" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient2" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient4" a demandé le film : Film5.
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient1".
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient3".
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient1"...
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient3"...
00 : 01 : 01: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo4" pour la demande du serveur "ServeurClient2".
00 : 01 : 02: Mise en relation du serveur video "ServeurVideo4" avec le serveur client "ServeurClient2"...
Transferts en cours :
ServeurVideo3->ServeurClient1 :
Film demandé : Film5
Qds demandée : 70 frames/sec
Reçu : 66 frames/sec.
Nbre de Frames restantes : 340 frames.
ServeurVideo3->ServeurClient3 :
Film demandé : Film5
Qds demandée : 35 frames/sec
Reçu : 31 frames/sec.
Nbre de Frames restantes : 340 frames.

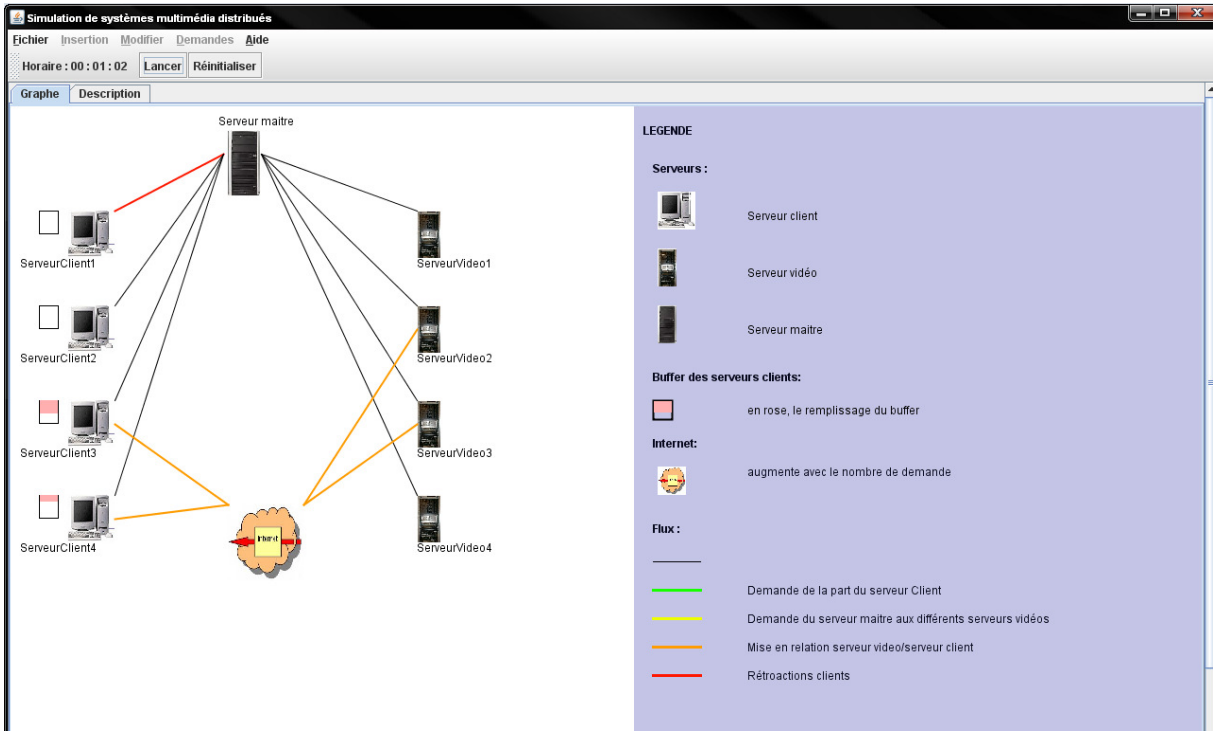
5ème étape : les chargements se terminent avec succès. Le serveur client 4 a du attendre avant de commencer son chargement.


Simulation de systèmes multimédia distribués
Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 01 : 48
Lancer
Réinitialiser
Graphe
Description
Evenements :
00 : 00 : 57: Le serveur Client "ServeurClient1" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient3" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient2" a demandé le film : Film5.
00 : 00 : 57: Le serveur Client "ServeurClient4" a demandé le film : Film5.
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient1".
00 : 01 : 00: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient3".
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient1"...
00 : 01 : 01: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient3"...
00 : 01 : 01: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo4" pour la demande du serveur "ServeurClient2".
00 : 01 : 02: Mise en relation du serveur video "ServeurVideo4" avec le serveur client "ServeurClient2"...
00 : 01 : 17: Terminaison avec succès du serveur "ServeurClient1".
00 : 01 : 17: Terminaison avec succès du serveur "ServeurClient3".
00 : 01 : 18: Terminaison avec succès du serveur "ServeurClient2".
00 : 01 : 20: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient4".
00 : 01 : 21: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient4"...
00 : 01 : 37: Terminaison avec succès du serveur "ServeurClient4".

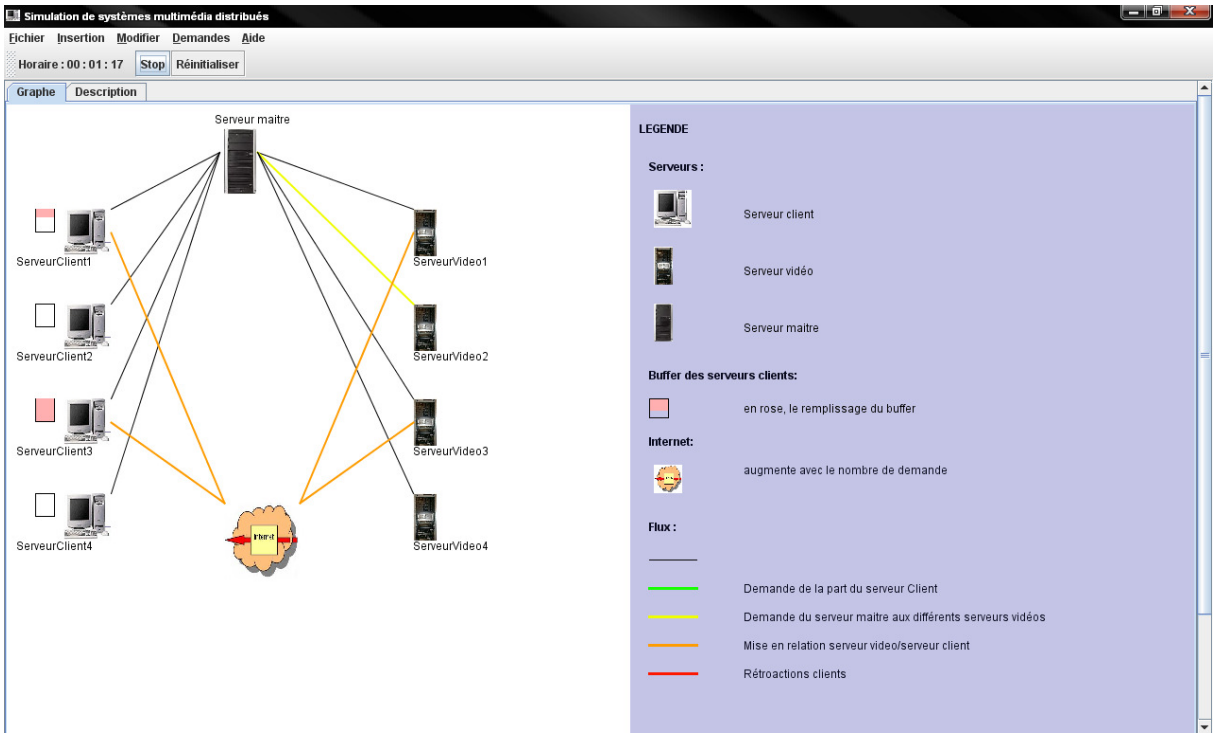
Exemple 2 : On effectue trois demandes dont une avec une faible QdS demandée.

Création de demandes simultanées
Serveur demandeur ServeurClient1 Film demandé : Film1 Qualite de service demandée : 22 frame/sec.
Serveur demandeur ServeurClient4 Film demandé : Film4 Qualite de service demandée : 25 frame/sec.
Serveur demandeur ServeurClient3 Film demandé : Film6 Qualite de service demandée : 35 frame/sec.
Ok
Annuler

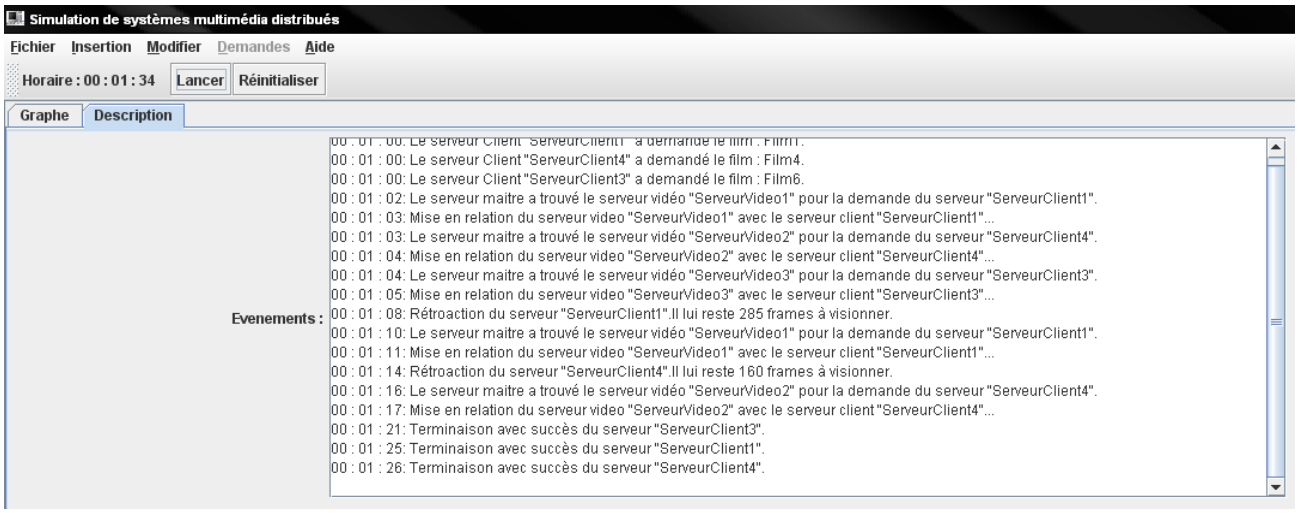
Après les différentes étapes, on peut voir que le serveur client 1 effectue une rétroaction.



Puis le serveur client 4 dû à la congestion du réseau. Le serveur maître commencera par interroger l'ancien serveur vidéo qui répondait à sa demande pour la rétroaction de celui-ci.



On peut apprécier grâce au panneau des événements que cette simulation se termine aussi avec succès :


Simulation de systèmes multimédia distribués
Fichier Insertion Modifier Demandes Aide
Horaire : 00 : 01 : 34
Lancer
Réinitialiser
Graphe
Description
Evenements :
00 : 01 : 00: Le serveur Client "ServeurClient4" a demandé le film : Film4.
00 : 01 : 00: Le serveur Client "ServeurClient3" a demandé le film : Film6.
00 : 01 : 02: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1" pour la demande du serveur "ServeurClient1".
00 : 01 : 03: Mise en relation du serveur video "ServeurVideo1" avec le serveur client "ServeurClient1"...
00 : 01 : 03: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo2" pour la demande du serveur "ServeurClient4".
00 : 01 : 04: Mise en relation du serveur video "ServeurVideo2" avec le serveur client "ServeurClient4"...
00 : 01 : 04: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo3" pour la demande du serveur "ServeurClient3".
00 : 01 : 05: Mise en relation du serveur video "ServeurVideo3" avec le serveur client "ServeurClient3"...
00 : 01 : 08: Rétroaction du serveur "ServeurClient1".Il lui reste 285 frames à visionner.
00 : 01 : 10: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1" pour la demande du serveur "ServeurClient1".
00 : 01 : 11: Mise en relation du serveur video "ServeurVideo1" avec le serveur client "ServeurClient1"...
00 : 01 : 14: Rétroaction du serveur "ServeurClient4".Il lui reste 160 frames à visionner.
00 : 01 : 16: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo2" pour la demande du serveur "ServeurClient4".
00 : 01 : 17: Mise en relation du serveur video "ServeurVideo2" avec le serveur client "ServeurClient4"...
00 : 01 : 21: Terminaison avec succès du serveur "ServeurClient3".
00 : 01 : 25: Terminaison avec succès du serveur "ServeurClient1".
00 : 01 : 26: Terminaison avec succès du serveur "ServeurClient4".

4.2. Améliorations

Le programme fonctionne correctement mais de nombreuses possibilités s'offrent pour son amélioration :

- Tout d'abord, il serait possible d'étendre le nombre de serveurs du système. Pour cette simulation, le nombre de serveur est limité pour le moment du à l'interface graphique.
- Etablir des horaires pour la congestion du réseau internet ou trouver un autre moyen de la simuler. Pour le moment, la congestion est simulée avec le nombre de demandes des clients.
- Lorsqu'un serveur client ne peut pas être servi, il attend tant que le serveur maître ne trouve pas de serveur vidéo disponible. Il faudrait gérer une rétroaction même s'il n'y a pas encore eu de transfert de données, ce qui sera peut-être effectué d'ici la soutenance de projet ou bien demandé à un serveur vidéo de transférer le film vers un autre serveur vidéo.
- La possibilité de défaillance d'un serveur vidéo n'est pour l'instant pas encore prise en charge ou tout du moins pas encore simulée.

ANNEXES

Les deux fichiers présentés ont été faits à partir de la même simulation.

Annexe 1 : Exemple de fichier .sim

```
0 0 53 53
4
ServeurClient1 120 0 0 0
ServeurClient3 110 45 0 0
ServeurClient4 120 0 0 0
ServeurClient5 110 110 0 0
4
ServeurVideo1 120 4 47 0
Film1 360
Film2 340
Film3 350
film7 160
ServeurVideo2 130 3 130 0
Film2 340
Film3 350
Film4 360
ServeurVideo4 150 2 150 0
Film1 360
Film5 340
ServeurVideo5 160 2 160 0
Pile 370
Face 400
3
ServeurClient3 39 Film1 16 335 0
1
1
0
1
42
ServeurClient5 39 Film3 47 125 0
0
1
0
1
42
ServeurClient3 46 Film1 26 260 1
0
1
0
0
1
48
00 : 00 : 07: Le serveur Client "ServeurClient1" a demandé le film : Face.
00 : 00 : 11: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo5"
pour la demande du serveur "ServeurClient1".
00 : 00 : 12: Mise en relation du serveur video "ServeurVideo5" avec le
serveur client "ServeurClient1"...
00 : 00 : 30: Terminaison avec succès du serveur "ServeurClient1".
00 : 00 : 39: Le serveur Client "ServeurClient3" a demandé le film : Film1.
00 : 00 : 39: Le serveur Client "ServeurClient5" a demandé le film : Film3.
00 : 00 : 40: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1"
pour la demande du serveur "ServeurClient3".
```

```
00 : 00 : 40: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1"
pour la demande du serveur "ServeurClient5".
00 : 00 : 41: Mise en relation du serveur video "ServeurVideo1" avec le
serveur client "ServeurClient3"...
00 : 00 : 41: Mise en relation du serveur video "ServeurVideo1" avec le
serveur client "ServeurClient5"...
00 : 00 : 44: Rétroaction du serveur "ServeurClient3".Il lui reste 335
frames à visionner.
00 : 00 : 46: Le serveur maitre a trouvé le serveur vidéo "ServeurVideo1"
pour la demande du serveur "ServeurClient3".
00 : 00 : 47: Mise en relation du serveur video "ServeurVideo1" avec le
serveur client "ServeurClient3"...
sauv
tps 0
sc
ServeurClient5 110
tps 0
supSC
1
tps 0
sv
ServeurVideo5 160 2
Pile 370
Face 400
tps 0
supSV
2
tps 0
modTB
1 110
tps 0
AjF
0 film7 160
tps 0
SuppF
2 1
tps 0
modQDS
1 130
tps 7
dem
0 7 Face 110
tps 39
dem
1 39 Film1 16
tps 39
dem
3 39 Film3 47
```

Annexe 2 : Exemple de fichier .vis

```
tps 0
sc
ServeurClient5 110
tps 0
supSC
1
tps 0
sv
ServeurVideo5 160 2
Pile 370
Face 400
tps 0
supSV
2
tps 0
modTB
1 110
tps 0
AjF
0 film7 160
tps 0
SuppF
2 1
tps 0
modQDS
1 130
tps 7
dem
0 7 Face 110
tps 39
dem
1 39 Film1 16
tps 39
dem
3 39 Film3 47
tps 53
fin
53
```